Solaris 可调参数参考手册



Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054
U.S.A.

文件号码 819–7063–10
2006 年 10 月

061016@15490

目录

前言

《*Solaris* **可调参数参考手册**》提供了有关 Solaris™ OS 内核和网络可调参数的参考信息。本手册并未提供有关 GNOME 或 Java™ 环境的可调参数信息。

本手册包含基于 SPARC® 和基于 x86 的系统的信息。

注 – 此 Solaris 发行版支持使用以下 SPARC 和 x86 系列处理器体系结构的系统：UltraSPARC ®、SPARC64、AMD64、Pentium 和 Xeon EM64T。支持的系统可以在 http://www.sun.com/bigadmin/hcl 上的《Solaris 10 Hardware Compatibility List》中找到。本文档列举了在不同类型的平台上进行实现时的所有差别。

在本文档中，这些与 x86 相关的术语表示以下含义：

- "x86" 泛指 64 位和 32 位的 x86 兼容产品系列。
- "x64" 指出了有关 AMD64 或 EM64T 系统的特定 64 位信息。
- “32 位 x86”指出了有关基于 x86 的系统的特定 32 位信息。

支持的系统可以在 http://www.sun.com/bigadmin/hcl 上的《Solaris 10 Hardware Compatibility List》中找到。

目标读者

本书适用于可能需要在某些情况下更改内核可调参数的有经验的 Solaris 系统管理员。有关更改 Solaris 可调参数的指导，请参阅第 22 页中的 “调优 Solaris 系统” 。

本书的结构

下表概述了本书中各章节和附录的内容。

章节	说明
第 1 章	Solaris 系统调优概述。此外，还介绍了本书中使用的描述内核可调参数的格式。

章节	说明
第 2 章	介绍 Solaris 内核可调参数，如内核内存、文件系统、进程大小以及分页参数。
第 3 章	介绍 NFS 可调参数，如高速缓存符号链接、动态重新传输以及 RPC 安全参数。
第 4 章	介绍 TCP/IP 可调参数，如 IP 转发、源路由以及缓存区大小调整参数。
第 5 章	介绍网络高速缓存和加速器 (Network Cache and Accelerator, NCA) 的可调参数。
第 6 章	介绍用于设置某些系统功能的缺省值的参数。通过修改 `/etc/default` 目录中的文件进行更改。
附录 A	已经更改或者现在已过时的参数的历史记录。
附录 B	包括当前 Solaris 发行版在内的本手册修订版的历史记录。

相关书籍

以下书籍提供了调优 Solaris 系统时可能使用的背景材料。

- 《Configuration and Capacity Planning for Solaris Servers》， Brian L. Wong 著， Sun Microsystems Press 出版， ISBN 0-13-349952–9
- 《NFS Illustrated》， Brent Callaghan 著， Addison Wesley 出版， ISBN 0-201-32570-5
- 《Resource Management》， Richard McDougall、 Adrian Cockcroft、 Evert Hoogendoorn、 Enrique Vargas、 Tom Bialaski 合著， Sun Microsystems Press 出版， ISBN 0-13-025855-5
- 《Sun Performance and Tuning:SPARC and Solaris》， Adrian Cockcroft 著， Sun Microsystems Press/PRT Prentice Hall 出版， ISBN 0-13-149642-3

有关 Solaris 调优信息的其他资源

此表介绍了 Solaris 调优信息的其他资源。

调优资源	更多信息
性能调优类	http://suned.sun.com
联机性能调优信息	http://www.sun.com/sun-on-net/performance
Sun Microsystems Press 出版的排序性能调优文档	http://www.sun.com/books/blueprints.series.html

文档、支持和培训

Sun Web 站点提供有关以下附加资源的信息：

- 文档 (http://www.sun.com/documentation/)
- 支持 (http://www.sun.com/support/)
- 培训 (http://www.sun.com/training/)

印刷约定

下表介绍了本书中的印刷约定。

表 P–1 印刷约定

字体	含义	示例
`AaBbCc123`	命令、文件和目录的名称；计算机屏幕输出	编辑 `.login` 文件。 使用 `ls -a` 列出所有文件。 `machine_name% you have mail.`
`AaBbCc123`	用户键入的内容，与计算机屏幕输出的显示不同	`machine_name%` **`su`** `Password:`
aabbcc123	要使用实名或值替换的命令行占位符	删除文件的命令为 `rm` *filename*。
AaBbCc123	保留未译的新词或术语以及要强调的词	这些称为 *Class* 选项。 **注意**：有些强调的项目在联机时以粗体显示。
新词术语强调	新词或术语以及要强调的词	**高速缓存**是存储在本地的副本。 请**勿**保存文件。
《书名》	书名	阅读《用户指南》的第 6 章。

命令中的 shell 提示符示例

下表列出了 C shell、Bourne shell 和 Korn shell 的缺省 UNIX® 系统提示符和超级用户提示符。

表 P–2 Shell 提示符

Shell	提示符
C shell	`machine_name%`

表 P–2 Shell 提示符　（续）

Shell	提示符
C shell 超级用户	`machine_name#`
Bourne shell 和 Korn shell	`$`
Bourne shell 和 Korn shell 超级用户	`#`

第 1 章

Solaris 系统调优概述

本节概述了本手册中调优信息的格式。另外，还介绍了调优 Solaris 系统的不同方法。

Solaris 10 6/06 发行版中 Solaris 系统调优方面的新增功能

本节介绍了 Solaris 10 6/06 发行版中新增或已更改的参数。

- ip_multidata_outbound 参数已得到增强。有关更多信息，请参见第 116 页中的 "ip_multidata_outbound" 。
- ip_squeue_fanout 参数已修改。有关更多信息，请参见第 118 页中的 "ip_squeue_fanout" 和第 118 页中的 "ip_soft_rings_cnt" 中的新增参数。

Solaris 10 发行版中对以下参数进行了更改，但是以前未记录这些更改。

- ip_forward_src_routed 和 ip6_forward_src_routed 参数已更正。从 Solaris 9 发行版开始，此参数的缺省值为禁用而不是启用。有关更多信息，请参见第 115 页中的 "ip_forward_src_routed 和 ip6_forward_src_routed" 。
- ip_squeue_write 参数名称已更改为 ip_squeue_enter。有关更多信息，请参见第 117 页中的 "ip_squeue_enter" 。
- logevent_max_q_sz 参数的缺省值已从 2000 个事件更改为 5000 个事件。有关更多信息，请参见第 32 页中的 "logevent_max_q_sz" 。
- 在 Solaris 10 发行版中，错误地记录了 lwp_default_stksize 参数。对于 SPARC 系统，其缺省值为 24,576。有关更多信息，请参见第 31 页中的 "lwp_default_stksize" 。

- 在 Solaris 10 发行版中，错误地记录了 sq_max_size 参数的缺省值。有关更多信息，请参见第 146 页中的 “sq_max_size” 。
- UDP 参数已更正。Solaris 10 发行版中对这些参数的缺省值进行了更改，但是以前未记录新的缺省值。有关更多信息，请参见第 132 页中的 “UDP 可调参数” 。

Solaris 10 发行版中 Solaris 系统调优方面的新增功能

本节介绍了 Solaris 10 发行版中新增或已更改的参数。

缺省栈大小

新增参数 default_stksize 可指定所有线程、内核或用户的缺省栈大小。lwp_default_stksize 参数仍然可用，但是不会影响所有内核栈。如果设置了 default_stksize，则会覆盖 lwp_default_stksize。有关更多信息，请参见第 30 页中的 “default_stksize” 。

System V IPC 配置

在本 Solaris 发行版中，所有 System V IPC 功能或者是自动配置的，或者可通过资源控制来控制。可共享的功能包括内存、消息队列以及信号。

通过资源控制，可以在本地系统或名称服务环境中按项目或按用户来进行 IPC 设置。

在以前的 Solaris 发行版中，IPC 功能由内核可调参数来控制。要更改这些功能的缺省值，必须修改 /etc/system 文件并重新引导系统。

由于 IPC 功能现在通过资源控制来控制，因此可以在系统运行过程中修改其配置。

许多以前需要系统调优才能正常运行的应用程序现在无需调优即可运行，因为增加了缺省资源并且资源是自动分配的。

下表介绍了现已过时的 IPC 可调参数及其替代资源控制。

资源控制	过时的可调参数	原有的缺省值	最大值	新缺省值
process.max-msg-qbytes	msginfo_msgmnb	4096	ULONG_MAX	65536
process.max-msg-messages	msginfo_msgtql	40	UINT_MAX	8192
process.max-sem-ops	seminfo_semopm	10	INT_MAX	512
process.max-sem-nsems	seminfo_semmsl	25	SHRT_MAX	512
project.max-shm-memory	shminfo_shmmax	0x800000	UINT64_MAX	物理内存的 1/4
project.max-shm-ids	shminfo_shmmni	100	2^{24}	128
project.max-msg-ids	msginfo_msgmni	50	2^{24}	128
project.max-sem-ids	seminfo_semmni	10	2^{24}	128

过时的参数仍可包括在 Solaris 系统上的 /etc/system 文件中。如果是这样，则这些参数可像在以前的 Solaris 发行版中一样，用来初始化缺省资源控制值。有关更多信息，请参见第 165 页中的 “过时或已删除的参数”。但是，建议不要使用过时的参数。

以下相关参数已被删除。如果这些参数包括在 Solaris 系统上的 /etc/system 文件中，则会注释掉这些参数。

semsys:seminfo_semmns	semsys:seminfo_semvmx
semsys:seminfo_semmnu	semsys:seminfo_semaem
semsys:seminfo_semume	semsys:seminfo_semusz
semsys:seminfo_semmap	shmsys:shminfo_shmseg
shmsys:shminfo_shmmin	msgsys:msginfo_msgmap
msgsys:msginfo_msgseg	msgsys:msginfo_msgssz
msgsys:msginfo_msgmax	

有关可用资源控制的当前列表，请参见 rctladm(1M)。有关配置资源控制的信息，请参见 project(4) 以及《System Administration Guide: Solaris Containers-Resource Management and Solaris Zones》中的第 6 章，“Resource Controls (Overview)”。

NFSv4 参数

本发行版中包括以下用于 NFSv4 协议的参数：

- 第 82 页中的 “nfs:nfs4_pathconf_disable_cache”
- 第 84 页中的 “nfs:nfs4_cots_timeo”

- 第 86 页中的 “nfs:nfs4_do_symlink_cache”
- 第 87 页中的 “nfs:nfs4_dynamic”
- 第 89 页中的 “nfs:nfs4_lookup_neg_cache”
- 第 91 页中的 “nfs:nfs4_max_threads”
- 第 93 页中的 “nfs:nfs4_nra”
- 第 95 页中的 “nfs:nfs4_shrinkreaddir”
- 第 99 页中的 “nfs:nfs4_bsize”
- 第 101 页中的 “nfs:nfs4_async_clusters”
- 第 104 页中的 “nfs:nfs4_max_transfer_size”

有关 NFSv4 参数的信息，请参见第 82 页中的 “NFS 模块参数” 。

新增和已更改的 TCP/IP 参数

本 Solaris 发行版中新增了以下 IP 参数：

- 第 117 页中的 “ip_squeue_worker_wait”
- 第 117 页中的 “ip_squeue_enter”
- 第 118 页中的 “ip_squeue_fanout”
- 第 129 页中的 “ipcl_conn_hash_size”

本 Solaris 发行版中新增了以下 TCP 参数：

- 第 127 页中的 “tcp_rst_sent_rate_enabled”
- 第 128 页中的 “tcp_rst_sent_rate”
- 第 128 页中的 “tcp_mdt_max_pbufs”

本 Solaris 发行版中，以下 TCP/IP 参数已过时：

- ipc_tcp_conn_hash_size
- tcp_compression_enabled
- tcp_conn_hash_size
- ip_forwarding
- ip6_forwarding
- xxx_forwarding

IP 转发更改

在本 Solaris 发行版中，使用 routeadm 命令或 ifconfig 命令来启用或禁用 IP 转发，而不是使用 ndd 命令设置以下可调参数来启用或禁用：

- ip_forwarding
- ip6_forwarding
- xxx_forwarding

使用 routeadm 命令和 ifconfig 命令而不是 ndd 命令来设置 IP 转发具有以下优点：

- 每次重新引导之后都会保留所有设置

- 新增的 `ifconfig` `router` 和 `-router` 命令可以与初始配置接口时运行的其他 `ifconfig` 命令一起放在 `/etc/hostname.`*interface* 文件中。

要在系统的所有接口上启用 IPv4 或 IPv6 包转发，可使用以下命令：

```
# routeadm -e ipv4-forwarding
```

```
# routeadm -e ipv6-forwarding
```

要在系统的所有接口上禁用 IPv4 或 IPv6 包转发，可使用以下命令：

```
# routeadm -d ipv4-forwarding
```

```
# routeadm -d ipv6-forwarding
```

在以前的 Solaris 发行版中，可使用如下命令在系统的所有接口上启用 IPv4 或 IPv6 包转发：

```
# ndd -set /dev/ip ip_forwarding 1
```

```
# ndd -set /dev/ip ip6_forwarding 1
```

在以前的 Solaris 发行版中，可使用如下命令在系统的所有接口上禁用 IPv4 或 IPv6 包转发：

```
# ndd -set /dev/ip ip_forwarding 0
```

```
# ndd -set /dev/ip ip6_forwarding 0
```

如果要在特定的 IPv4 接口或 IPv6 接口上启用 IP 转发，可针对相应接口使用以下类似语法。以下以 `bge0` 接口为例。

```
# ifconfig bge0 router
```

```
# ifconfig bge0 inet6 router
```

如果要在特定的 IPv4 接口或 IPv6 接口上禁用 IP 转发，可针对相应接口使用以下类似语法。以下以 `bge0` 接口为例。

```
# ifconfig bge0 -router
```

```
# ifconfig bge0 inet6 -router
```

以前，使用如下命令在特定接口上启用 IP 转发：

```
# ndd -set /dev/ip bge0:ip_forwarding 1
```

```
# ndd -set /dev/ip bge0:ip_forwarding 1
```

以前，使用如下命令在特定接口上禁用 IP 转发：

```
# ndd -set /dev/ip ip_forwarding 0
```

```
# ndd -set /dev/ip ip6_forwarding 0
```

如果希望上述任一 `routeadm` 设置在运行的系统上生效，请使用以下命令：

```
# routeadm -u
```

有关更多信息，请参见 `routeadm`(1M) 和 `ifconfig`(1M)。

SPARC: 转换存储缓冲区 (Translation Storage Buffer, TSB) 参数

本发行版中包括用于调优转换存储缓冲区 (Translation Storage Buffer, TSB) 的新增参数。有关 TSB 参数的信息，请参见第 74 页中的 "Sun-4u 特定参数"。

SCTP 可调参数

本 Solaris 发行版中提供了流控制传输协议 (Stream Control Transmission Protocol, SCTP)，它是一种可靠的传输协议，提供的服务与 TCP 所提供的类似。有关 SCTP 可调参数的更多信息，请参见第 133 页中的 "SCTP 可调参数"。

调优 Solaris 系统

Solaris OS 是在 SPARC 和 x86 处理器上运行的多线程且可伸缩的 UNIX® 操作系统。它可根据系统负载自行调整，并且只需进行最少的调优。但是，在某些情况下，必须进行调优。本书提供了有关可用于 Solaris OS 的正式支持的内核调优选项的详细信息。

Solaris 内核包括一个始终装入的核心部分，以及一些只在被引用时才装入的可装入模块。本指南的内核部分中引用的许多变量都位于核心部分。但是，一些变量则位于可装入模块中。

系统调优中的一个关键注意事项是：设置系统参数（或系统变量）通常是可用来提高性能的效果最差的操作，而更改应用程序的行为通常可以最有效地帮助实现系统调优。添加更多物理内存以及平衡磁盘 I/O 模式也非常有用。在极少数情况下，更改本指南中介绍的其中一个变量将对系统性能产生很大影响。

请记住，一个系统的 `/etc/system` 设置，无论是从整体还是部分而言，可能都不适用于另一个系统的环境。请针对应用该文件中的设置值的环境仔细考虑这些值。请确保首先了解一个系统的行为，然后再尝试对此处介绍的系统变量应用更改。



注意 – 本书中介绍的可调参数在不同的发行版之间可以有所不同，实际情况也是如此。发行版可以是 Solaris Update 发行版，也可以是新的发行版，如 Solaris 10。发布这些可调参数时，不排除在未发出通知的情况下对可调参数及其说明进行的更改。

调优可调参数时应用的说明格式

对于每个可调参数，其说明格式如下：

- 参数名称
- 说明
- 数据类型
- 缺省值
- 范围
- 单位
- 是否为动态
- 验证
- 含义
- 何时更改
- 承诺级别
- 更改历史记录

参数名称

`/etc/system` 文件中键入或在 `/etc/default/`*facility* 文件中找到的确切名称。

大多数参数名称的形式为不包含冒号 (`:`) 的**参数**。这些名称是指内核的核心部分中的变量。如果名称确实包含冒号，则冒号左侧的字符将引用可装入模块的名称。模块内参数的名称由冒号右侧的字符组成。例如：

module_name:*variable*

说明

简要介绍参数的作用及其控制的内容。

数据类型

指示带符号或无符号的短整数或长整数，区别如下：

- 在运行 32 位内核的系统上，长整数与整数大小相同。
- 在运行 64 位内核的系统上，长整数的宽度（以位为单位）是整数的两倍。例如，无符号整数 = 32 位，无符号长整数 = 64 位。

单位

（可选）介绍单位类型。

缺省值

系统用作缺省值的值。

范围

指定系统验证允许的可能范围或数据类型的限制。

- MAXINT－带符号整数的最大值 (2,147,483,647) 的缩略说明
- MAXUINT－无符号整数的最大值 (4,294,967,295) 的缩略说明

是否为动态	如果可以使用 mdb 或 kmdb 调试程序在运行的系统上更改参数，则表明此参数为动态参数。如果参数只能在引导时进行初始化，则表明此参数不是动态参数。
验证	检查系统是应用 /etc/system 文件中指定的变量值还是应用缺省值，并检查何时应用验证。
含义	（可选）提供可能针对此参数存在的未声明的约束，尤其是相对于其他参数的约束。
何时更改	说明某个用户可能要更改该值的原因，其中包括错误消息或返回代码。
承诺级别	标识接口的稳定性。本手册中的许多参数仍在改进中，因此属于不稳定参数。有关更多信息，请参见attributes(5)。
更改历史记录	（可选）包含指向“更改历史记录”附录的链接（如果适用）。

调优 Solaris 内核

下表介绍了可以应用可调参数的不同方法。

应用可调参数的方法	更多信息
修改 /etc/system 文件	第 24 页中的 “/etc/system 文件”
使用内核调试程序 (kmdb)	第 25 页中的 “kmdb 命令”
使用模块调试程序 (mdb)	第 25 页中的 “mdb 命令”
使用 ndd 命令设置 TCP/IP 参数	第 4 章
修改 /etc/default 文件	第 143 页中的 “调优 NCA 参数”

/etc/system 文件

/etc/system 文件提供了用于调整内核参数值的静态机制。引导时会读取此文件中指定的值并应用这些值。对此文件所做的任何更改仅在重新引导操作系统之后才会应用于该系统。

在 Solaris 8 发行版之前，在以下两个阶段中应用了用于设置参数值的 /etc/system 项：

- 第一个阶段将获取各种引导参数（例如 maxusers）来初始化关键的系统参数。
- 第二个阶段将使用引导参数来计算基本配置，并应用 /etc/system 文件中指定的所有值。对于引导参数，重新应用的值将替换在初始化阶段中计算或重置的值。

有时，第二个阶段会对用户和管理员造成困惑，原因是将参数设置为似乎不允许的值，或是对初始配置过程中覆盖了其中一个值的参数（如 max_nprocs）进行了赋值。

从 Solaris 8 发行版开始，在计算配置参数之前将一次设置所有值。

示例—在 /etc/system 中设置参数

以下 /etc/system 项用于设置针对使用 NFS 版本 2 软件挂载的文件系统所读取的读前块数。

```
set nfs:nfs_nra=4
```

从错误值中恢复

请在修改 /etc/system 文件之前生成其副本，以便可以轻松从错误值中恢复。例如：

```
# cp /etc/system /etc/system.good
```

如果 /etc/system 文件中指定的值导致系统无法引导，则可以使用以下命令进行恢复：

```
ok boot -a
```

此命令将使系统要求提供在引导过程中使用的各种文件的名称。请按回车键接受缺省值，直到请求 /etc/system 文件的名称为止。显示 Name of system file [/etc/system]: 提示时，请键入完好的 /etc/system 文件或 /dev/null 的名称：

```
Name of system file [/etc/system]: /etc/system.good
```

如果指定了 /dev/null，则此路径将导致系统尝试从 /dev/null 中读取其配置信息。由于此文件为空，因此系统会使用缺省值。引导系统之后，即可更正 /etc/system 文件。

有关系统恢复的更多信息，请参见《System Administration Guide: Basic Administration》。

kmdb 命令

kmdb 是一个交互式内核调试程序，其通用语法与 mdb 相同。交互式内核调试程序的一个优点是可以设置断点。到达断点时，可以通过执行内核代码来检查数据或步骤。

可以根据需要装入和卸载 kmdb。不必像在 kadb 中那样重新引导系统来执行交互式内核调试。

有关更多信息，请参见kmdb(1)。

mdb 命令

从 Solaris 8 发行版开始，将使用 mdb 作为模块调试程序，这是一种独特的 Solaris 调试程序，因为它很容易扩展。其中还有一个编程 API，可用于对模块进行编译，以便在调试程序环境中执行所需任务。

mdb 还包括一些所需的可用性功能，如命令行编辑、命令历史记录、内置输出页面调度程序、语法检查以及命令管道传输。mdb 是建议针对内核使用的事后调试程序。

有关更多信息，请参见mdb(1)。

示例－使用 mdb 更改值

要将整数参数 maxusers 的值从 495 更改为 512，请执行以下操作：

```
# mdb -kw

Loading modules: [ unix krtld genunix ip logindmux ptm nfs ipc lofs ]

> maxusers/D

maxusers:

maxusers:       495

> maxusers/W 200

maxusers:       0x1ef           =       0x200

> $q
```

使用要更改的项的实际地址以及此参数将设置为的值来替换 maxusers。

有关使用模块调试程序的更多信息，请参见《Solaris 模块调试程序指南》。

使用 kmdb 或 mdb 调试程序时，不需要模块名称前缀。装入模块之后，其符号将形成一个公用名称空间，其中包括核心内核符号以及先前装入的其他任何模块符号。

例如，每个调试程序中会将 ufs:ufs_WRITES 作为 ufs_WRITES 进行访问（假定装入了 UFS 模块）。在 /etc/system 文件中进行设置时，需要 ufs: 前缀。

特殊的 **Solaris** tune 和 var 结构

Solaris 可调参数具有各种形式。tune 结构（在 /usr/include/sys/tuneable.h 文件中定义）是 tune_t_fsflushr、tune_t_minarmem 以及 tune_t_flkrec 的运行时表示形式。初始化内核之后，可在 tune 结构的相应字段中找到对这些变量的所有引用。

各种文档（例如《Solaris System Administration Guide, Volume 2》的早期版本）均已指明在 tune 结构中设置参数的正确方法是使用语法 tune:*field-name*，其中 *field-name* 将由以上列出的实际参数名称所替换。此过程将在不出现提示的情况下失败。在引导时针对此结构设置参数的正确方法是初始化对应于所需字段名称的特殊参数。然后，系统初始化过程会将这些值装入 tune 结构。

用来放置各种可调参数的另一个结构是名为 v 的 var 结构。可以在 /usr/include/sys/var.h 文件中找到 var 结构的定义。变量（如 autoup 和 bufhwm）的运行时表示形式即存储在此位置。

请勿在运行的系统上更改 tune 或 v 结构。在运行的系统上更改这些结构中的任何字段都可能导致系统发生混乱。

查看 Solaris 系统配置信息

可使用多种工具检查系统配置信息。部分工具需要超级用户权限。其他工具可以由非特权用户运行。通过在运行的系统上使用 mdb 或在 kmdb 下进行引导，可以借助内核调试程序来检查每个结构和数据项。

有关更多信息，请参见mdb(1) 或 kadb(1M)。

sysdef 命令

sysdef 命令可提供 System V IPC 设置、STREAMS 可调参数、进程资源限制以及 tune 和 v 结构的各部分的值。例如，512 MB 的 Sun™ Ultra™ 80 系统上的 sysdef "可调参数"部分如下所示：

```
10387456        maximum memory allowed in buffer cache (bufhwm)

    7930        maximum number of processes (v.v_proc)

      99        maximum global priority in sys class (MAXCLSYSPRI)

    7925        maximum processes per user id (v.v_maxup)

      30        auto update time limit in seconds (NAUTOUP)

      25        page stealing low water mark (GPGSLO)

       5        fsflush run rate (FSFLUSHR)

      25        minimum resident memory for avoiding deadlock (MINARMEM)

      25        minimum swapable memory for avoiding deadlock (MINASMEM)
```

有关更多信息，请参见sysdef(1M)。

kstat 实用程序

kstat 是由各种内核子系统和驱动程序维护的数据结构。它们提供了一种机制，可将内核中的数据导出到用户程序，而无需程序读取内核内存或具有超级用户权限。有关更多信息，请参见kstat(1M) 或 kstat(3KSTAT)。

从 Solaris 8 发行版开始，kstat 命令可用于通过命令行界面来选择和显示 kstat。另外，也可以使用 Perl 模块 Kstat(3PERL) 来处理 kstat 信息。

注 – unix 模块中名为 system_pages 的 kstat 数据结构不会报告 cachefree 的统计信息。从 Solaris 9 发行版开始，不支持 cachefree。

◆ ◆ ◆ 第 2 章

Solaris 内核可调参数

本章介绍大多数 Solaris 内核可调参数。

有关可调参数的参考信息

可调参数	参考
NFS 可调参数	第 3 章

可调参数	参考
Internet 协议套件可调参数	第 4 章
网络高速缓存和加速器 (Network Cache and Accelerator, NCA) 可调参数	第 5 章

常规内核参数和内存参数

本节介绍与物理内存和栈配置相关的常规内核参数。

physmem

说明	考虑 Solaris OS 和固件所占用的物理内存之后，修改系统的物理内存页数配置。
数据类型	无符号长整数
缺省值	系统上可用物理内存页数（未将存储核心内核和数据的内存计算在内）
范围	1 到系统上的物理内存量
单位	页
是否为动态	否
验证	无
何时更改	需要测试使用更少的物理内存运行系统的效果时。由于此参数**没有**考虑核心内核和数据使用的内存，也没有考虑先前在启动过程中分配的各种其他数据结构使用的内存，因此 physmem 的值应该小于表示较小内存量的实际页数。
承诺级别	不稳定

default_stksize

说明　指定所有线程的缺省栈大小。无法创建任何栈大小小于 default_stksize 的线程。如果设置了 default_stksize，则会覆盖 lwp_default_stksize。另请参见第 31 页中的 “lwp_default_stksize”。

数据类型　整数

缺省值

- 在 SPARC 系统上，为 PAGESIZE 的 3 倍
- 在 x86 系统上，为 PAGESIZE 的 2 倍
- 在 AMD64 系统上，为 PAGESIZE 的 5 倍

范围　最小值为缺省值：

- 在 SPARC 系统上，为 PAGESIZE 的 3 倍
- 在 x86 系统上，为 PAGESIZE 的 2 倍
- 在 AMD64 系统上，为 PAGESIZE 的 5 倍

最大值为缺省值的 32 倍。

单位　字节，大小为 getpagesize 参数返回的值的倍数。有关更多信息，请参见 getpagesize(3C)。

是否为动态　是。影响更改变量之后创建的线程。

验证　必须大于或等于 8192，并且小于或等于 262,144 (256 x 1024)。同时，还必须是系统页面大小的倍数。如果不满足这些条件，则会显示以下消息：

```
Illegal stack size, Using N
```

值 *N* 为 default_stksize 的缺省值。

何时更改　系统由于用完栈空间而发出警告音时。此问题的最佳解决方案是确定系统用完空间的原因，然后再进行更正。

增大缺省栈大小意味着几乎每个内核线程都将具有更大的栈，从而导致无故增加内核内存消耗。通常不会使用增加的空间。消耗增加意味着争用同一内存池的其他资源具有的可用空间量将减少，从而可能会降低系统执行操作的能力。其中一个负面影响是内核可创建的线程数减少。在找出根本原因并解决问题之前，只应将此解决方案作为一种临时解决方法。

承诺级别　不稳定

lwp_default_stksize

说明　指定栈大小的缺省值，在创建内核线程时，如果调用例程未提供要使用的确切栈大小，则将使用该值。

数据类型　整数

缺省值

- 对于 x86 平台，为 PAGESIZE 的 2 倍
- 对于 SPARC 平台，为 PAGESIZE 的 5 倍
- 对于 AMD64 平台，为 PAGESIZE 的 3 倍

范围　最小值为缺省值：

- 在 SPARC 系统上，为 PAGESIZE 的 3 倍
- 在 x86 系统上，为 PAGESIZE 的 2 倍
- 在 AMD64 系统上，为 PAGESIZE 的 5 倍

最大值为缺省值的 32 倍。

单位　字节，大小为 getpagesize 参数返回的值的倍数。有关更多信息，请参见 getpagesize(3C)。

是否为动态　是。影响更改变量之后创建的线程。

验证　必须大于或等于 8192，并且小于或等于 262,144 (256 x 1024)。同时，还必须是系统页面大小的倍数。如果不满足这些条件，则会显示以下消息：

```
Illegal stack size, Using N
```

值 *N* 为 lwp_default_stksize 的缺省值。

何时更改　系统由于用完栈空间而发出警告音时。此问题的最佳解决方案是确定系统用完空间的原因，然后再进行更正。

增大缺省栈大小意味着几乎每个内核线程都将具有更大的栈，从而导致无故增加内核内存消耗。通常不会使用增加的空间。消耗增加意味着争用同一内存池的其他资源具有的可用空间量将减少，从而可能会降低系统执行操作的能力。其中一个负面影响是内核可创建的线程数减少。在找出根本原因并解决问题之前，只应将此解决方案作为一种临时解决方法。

承诺级别　不稳定

更改历史记录　有关信息，请参见第 156 页中的 “lwp_default_stksize（Solaris 9 发行版）”。

logevent_max_q_sz

说明　允许排队等待传送到 sysevent d 守护进程的最大系统事件数。一旦系统事件队列的大小达到此限制，便不再允许向队列中添加任何其他系统事件。

数据类型　整数

缺省值　5000

范围　0 到 MAXINT

单位　系统事件

是否为动态　是

验证　每当 ddi_log_sysevent 和 sysevent_post_event 生成系统事件时，系统事件框架都会检查此值。

有关更多信息，请参见 ddi_log_sysevent(9F) 和 sysevent_post_event (3SYSEVENT)。

何时更改　错误日志消息指示无法记录、生成或发布系统事件时。

承诺级别	不稳定
更改历史记录	有关信息，请参见第 156 页中的 “logevent_max_q_sz（Solaris 9 发行版）”。

segkpsize

说明	指定可用的内核可分页内存量。此内存主要用于内核线程栈。增大此数值可允许相同数量的线程使用更大的栈或允许存在更多的线程。只能在运行 64 位内核的系统上设置此参数。运行 64 位内核的系统使用的缺省栈大小为 24 KB。
数据类型	无符号长整数
缺省值	对于 64 位内核，为 2 GB 对于 32 位内核，为 512 MB
范围	对于 64 位内核，为 512 MB 到 24 GB
单位	8 KB 页
是否为动态	否
验证	将此值与最小值和最大值（对于 64 位系统，分别为 512 MB 和 24 GB）进行比较。如果此值小于最小值或大于最大值，则将其重置为 2 GB，同时显示相关消息。 创建高速缓存时使用的实际大小是验证检查之后在 segkpsize 中指定的值和物理内存的 50% 之间的较小值。
何时更改	需要支持系统上的大量进程时。缺省大小为 2 GB（假设至少存在 1 GB 的物理内存）。此缺省大小允许为多于 87,000 个的内核线程创建 24 KB 的栈。在 64 位内核中，无论进程是 32 位进程还是 64 位进程，栈的大小都相同。如果所需大小超过此值，则可以增大 segkpsize（假设存在足够的物理内存）。
承诺级别	不稳定
更改历史记录	有关信息，请参见第 156 页中的 “segkpsize（Solaris 9 12/02 发行版）”。

noexec_user_stack

说明	用于将栈标记为不可执行，这样可阻止缓冲区溢出攻击。

缺省情况下，运行 64 位内核的 Solaris 系统将所有 64 位应用程序的栈都标记为不可执行。要使 32 位应用程序不能在运行 64 位或 32 位内核的系统上执行，则有必要设置此参数。

注 – 此参数存在于所有运行 Solaris 2.6、7、8、9 或 10 发行版的系统上，但它仅对 64 位 SPARC 和 AMD64 体系结构有效。

数据类型	带符号整数
缺省值	0（禁用）
范围	0（禁用）或 1（启用）
单位	切换（开/关）
是否为动态	是。不会影响当前正在运行的进程，只影响在设置此值之后创建的进程。
验证	无
何时更改	应始终处于启用状态，除非应用程序要故意在不使用 mprotect 的情况下在栈中放置可执行代码以使栈可执行。有关更多信息，请参见 mprotect(2)。
承诺级别	不稳定
更改历史记录	有关信息，请参见第 156 页中的 “noexec_user_stack（Solaris 9 发行版）”。

fsflush 及相关参数

本节介绍 fsflush 及相关可调参数。

fsflush

定期运行系统守护进程 fsflush 以执行以下三项主要任务：

1. 每次调用时，fsflush 都会将一段时间内的脏文件系统页刷新到磁盘。
2. 每次调用时，fsflush 都会检查部分内存并将已修改的页写入其后备存储。如果页已经过修改，并且不满足以下任一条件，则会写入页：
   - 页为内核页
   - 页为可用页
   - 页为锁定页
   - 页与交换设备关联
   - 页当前参与 I/O 操作

实际结果是刷新文件中使用具有写入权限的 mmap 进行映射的页以及实际已更改的页。

页会被刷新到后备存储，但是仍与使用它们的进程保持连接。当系统在低内存的情况下运行时，如果自刷新以来页未被修改过，则这种方式可以避免因先将页写入后备存储而后再声明页所造成的延迟，从而简化页的回收过程。

3. fsflush 将文件系统元数据写入磁盘。每进行 *n* 次调用便执行一次此写入操作，其中 *n* 根据各种配置变量计算得出。有关详细信息，请参见第 35 页中的 “tune_t_fsflushr”和第 35 页中的 “autoup”。

可以配置以下功能：

- 调用的频率 (tune_t_fsflushr)
- 是否执行内存扫描 (dopageflush)
- 是否进行文件系统数据刷新 (doiflush)
- 文件系统数据的刷新频率 (autoup)

对于大多数系统，内存扫描和文件系统元数据同步是 fsflush 的主要活动。根据系统的使用情况，内存扫描可能很少使用，否则会占用过多的 CPU 时间。

tune_t_fsflushr

说明	指定 fsflush 调用的间隔秒数
数据类型	带符号整数
缺省值	1
范围	1 到 MAXINT
单位	秒
是否为动态	否
验证	如果此值小于或等于零，则会将其重置为 1 并显示警告消息。仅在引导时执行此检查。
何时更改	请参见 autoup 参数。
承诺级别	不稳定

autoup

说明	autoup 与 tune_t_flushr 一起控制每次调用时检查其中脏页的内存量以及文件系统同步操作的频率。

autoup 的值还可用于控制是否将缓冲区从可用列表写出。每当带有 B_DELWRI 标志（标识已更改的文件内容页）的缓冲区在列表中的存在时间超过 *autoup* 秒时，便会将其写出。增大 autoup 的值可延长缓冲区在内存中的保留时间。

数据类型　带符号整数

缺省值　30

范围　1 到 MAXINT

单位　秒

是否为动态　否

验证　如果 autoup 小于或等于零，则会将其重置为 30 并显示警告消息。仅在引导时执行此检查。

含义　autoup 应该是 tune_t_fsflushr 的整数倍。autoup 至少应该是 tune_t_fsflushr 值的 6 倍。 否则，每次调用 fsflush 时，都会扫描过多的内存量。

系统总页数与 tune_t_fsflushr 的乘积应大于或等于 autoup，以便在 dopageflush 不为零时检查内存。

何时更改　在下列情况下，可能要更改 autoup 和/或 tune_t_fsflushr 参数：

- 内存较大的系统－在这种情况下，增大 autoup 会减小每次调用 fsflush 时扫描的内存量。
- 内存需求最低的系统－同时增大 autoup 和 tune_t_fsflushr 会减少扫描次数。同时增大 autoup 是为了维护当前的 autoup / tune_t_fsflushr 比率。
- 具有大量瞬态文件的系统（例如，邮件服务器或软件生成计算机）－如果创建了大量文件然后将其删除，则 fsflush 可能不必将这些文件的数据页写入磁盘。

承诺级别　不稳定

dopageflush

说明　控制是否在调用 fsflush 期间检查内存中已修改的页。每次调用 fsflush 时，都会确定系统中的内存页数。此数值可能会因动态重新配置操作而发生了更改。每次调用都使用以下算法进行扫描：总页数 x tune_t_fsflushr / autoup 页数

数据类型　带符号整数

缺省值　1（启用）

范围　0（禁用）或 1（启用）

单位	切换（开/关）
是否为动态	是
验证	无
何时更改	如果系统页面扫描程序很少运行（由 vmstat 输出的 sr 列中的 0 值指示）。
承诺级别	不稳定

doiflush

说明	控制是否在调用 fsflush 期间执行文件系统元数据同步。每进行 *N* 次 fsflush 调用便会执行一次此同步操作，其中 *N*= (autoup / tune_t_fsflushr)。由于此算法是整数除法，因此，如果 tune_t_fsflushr 大于 autoup，则每次调用 fsflush 时都会执行同步，因为代码会检查其重复计数器，看是否大于还是等于 *N*。请注意，每次调用 fsflush 时都会计算 *N* 一次。随后对 tune_t_fsflushr 或 autoup 所做的更改不会影响同步操作的频率。
数据类型	带符号整数
缺省值	1（启用）
范围	0（禁用）或 1（启用）
单位	切换（开/关）
是否为动态	是
验证	无
何时更改	一段时间内频繁修改文件并且刷新操作引起的负载扰乱了系统行为。 如果文件的存在性以及状态的一致性在重新引导系统时都无关紧要，则最好将这些文件保留在 TMPFS 文件系统（如 /tmp）中。从 Solaris 7 发行版开始，可以使用 mount -noatime 选项减少系统上的 inode 通信流量。此选项可避免在访问文件时进行 inode 更新。 对于参与实时处理的系统，可能需要禁用此选项并使用显式应用程序文件同步来实现一致性。
承诺级别	不稳定

进程大小调整参数

可使用几个参数（或变量）来控制系统上的可用进程数以及单个用户可以创建的进程数。基础参数为 maxusers。此参数确定指定给 max_nprocs 和 maxuprc 的值。

maxusers

说明 最初，maxusers 定义系统可以支持的登录用户数。生成内核时，将基于该设置调整各种表的大小。当前 Solaris 发行版会基于系统上的内存量执行大量大小调整操作。因此，maxusers 过去的许多用法都已发生更改。仍由 maxusers 派生的许多子参数包括：

- 系统上的最大进程数
- 系统中保留的配额结构数
- 目录名称查找高速缓存 (directory name look-up cache, DNLC) 的大小

数据类型 带符号整数

缺省值 内存量（以 MB 为单位）与 2048 之间的较小值

范围 1 到 2048（如果未在 /etc/system 文件中设置，则基于物理内存设置）

1 到 4096（如果在 /etc/system 文件中设置）

单位 用户

是否为动态 否。计算完相关参数之后，便不再引用 maxusers。

验证 无

何时更改 由系统派生的缺省用户进程数太少时。在这种情况下，系统控制台将显示以下消息：

```
out of processes
```

当缺省进程数太多时，也可能要更改此参数，例如以下情况：

- 减小 maxusers 的缺省值时，具有大量内存但运行的进程相对较少的数据库服务器可以节省系统内存。
- 如果文件服务器具有大量内存但运行的进程较少，则可能要减小此值。不过，应显式设置 DNLC 的大小。请参见第 57 页中的 “ncsize”。
- 如果计算服务器具有大量内存但运行的进程较少，则可能要减小此值。

承诺级别 不稳定

reserved_procs

说明	指定要在进程表中为具有超级用户 UID (0) 的进程保留的系统进程槽数。例如，fsflush 具有超级用户 UID (0)。
数据类型	带符号整数
缺省值	5
范围	5 到 MAXINT
单位	进程
是否为动态	否。计算完初始参数之后便不再使用。
验证	从 Solaris 8 发行版开始，不会忽略任何 /etc/system 设置。
承诺级别	不稳定
何时更改	考虑增加到 10 + 系统上正常 UID 0（超级用户）进程数。当需要获取根 shell，否则系统便无法创建用户级进程时，该设置可提供一定程度的缓解。

pidmax

说明	指定可能的最大进程 ID 值。对 Solaris 8 以及更高发行版有效。 pidmax 设置 maxpid 变量的值。设置 maxpid 之后，便会忽略 pidmax。maxpid 在内核中的其他位置使用，用于确定最大进程 ID 和进行验证检查。 任何通过向 /etc/system 文件中添加项以设置 maxpid 的尝试都将无效。
数据类型	带符号整数
缺省值	30,000
范围	266 到 999,999
单位	进程
是否为动态	否。仅在引导时用于设置 pidmax 的值。
验证	是。会将此值与 reserved_procs 的值和 999,999 进行比较。如果此值小于 reserved_procs 或大于 999,999，则将其设置为 999,999。
含义	检查 max_nprocs 范围可确保 max_nprocs 始终小于或等于此值。
何时更改	需要在系统上启用对多于 30,000 个进程的支持时。
承诺级别	不稳定

max_nprocs

说明 指定可在系统上创建的最大进程数。包括系统进程和用户进程。使用 /etc/system 中指定的任一值来计算 maxuprc。

此值还可用来确定几个其他系统数据结构的大小。此参数还将在以下其他数据结构中起作用：

- 确定目录名称查找高速缓存的大小（如果未指定 ncsize）
- 为 UFS 分配磁盘配额结构（如果未指定 ndquot）
- 检验已配置的 System V 信号所使用的内存量是否未超过系统限制
- 为 x86 平台配置硬件地址转换资源。

数据类型 带符号整数

缺省值 10 + (16 x maxusers)

范围 266 到 maxpid 的值

是否为动态 否

验证 是。将此值与 maxpid 进行比较，如果此值较大，则将其设置为 maxpid。在 x86 平台上，还会根据特定于平台的值进行其他检查。将 max_nprocs 设置为三个值（max_nprocs、maxpid、平台值）中的最小者。SPARC 和 x86 平台都使用 65,534 作为平台值。

何时更改 更改此参数是在系统上启用对多于 30,000 个进程的支持所必需的一步。

承诺级别 不稳定

更改历史记录 有关信息，请参见第 155 页中的 “max_nprocs（Solaris 9 发行版）”。

maxuprc

说明 指定任一用户可在系统上创建的最大进程数。

数据类型 带符号整数

缺省值 max_nprocs - reserved_procs

范围 1 到 max_nprocs - reserved_procs

单位 进程

是否为动态 否

验证 是。会将此值与 max_nprocs - reserved_procs 进行比较，并将其设置为这两个值中的较小者。

何时更改 需要对用户可创建的进程数指定硬限制，使其小于系统可创建进程数的缺省值时。尝试超过此限制时，会在控制台上或消息文件中生成以下警告消息：

`out of per-user processes for uid` *N*

承诺级别	不稳定

与分页相关的参数

Solaris OS 使用按需分页的虚拟内存系统。系统运行时，会根据需要将页放入内存。当占用的内存超过特定阈值并且内存需求不断增加时，便会开始分页。可以通过几个级别进行分页，这些级别由特定参数控制。

常规分页算法如下：

- 发现内存不足。页面扫描程序线程将运行并开始扫描内存。将采用两步算法：
  1. 将页标记为未使用。
  2. 如果在一段时间间隔之后仍未使用，则将此页视为回收对象。

  如果该页已经过修改，则会向页出线程发出调度该页以执行 I/O 的请求。此外，页面扫描程序会继续查看内存。页出会导致将该页写入其后备存储并放入可用列表中。页面扫描程序扫描内存时，不会对页的来源进行区分。页可能来自数据文件，也可能来自可执行文件的文本、数据或栈。
- 随着系统内存压力的增大，此算法会越来越主动地调整将被它视为回收候选对象的页的数目以及分页算法运行的频率。（有关更多信息，请参见第 46 页中的 “fastscan”和第 47 页中的 “slowscan”。）随着可用内存从 `lotsfree` 降到 `minfree`，系统会将每次调用页出线程时扫描的内存量从 `slowscan` 指定的值线性增大到 `fastscan` 指定的值。系统使用 `desfree` 参数来控制许多有关资源使用和行为的决策。

最初，系统限定自身针对 `pageout` 操作所使用的 CPU 百分比不超过 4%。随着内存压力的增大，为支持 `pageout` 操作而占用的 CPU 时间会线性增大，直到最多占用 CPU 的 80% 为止。此算法将扫描 `slowscan` 和 `fastscan` 之间的某一内存量，然后在出现下列任一情况时停止扫描：

- 已找到足以解决内存不足问题的页。
- 已查看了计划的页数。
- 使用了过多的时间。

如果页出线程完成其扫描时仍存在内存不足问题，则在接下来的 1/4 秒时间内将安排其他扫描。

从 Solaris 9 发行版开始，更改了分页子系统的配置机制。系统将在引导时确定 `fastscan`、`slowscan` 和 `handspreadpages` 参数的相应设置，而不是依赖于这些参数的一组预定义值。在 `/etc/system` 文件中设置这些参数中的任何一个都会导致系统使用非最优值。



注意 – 请从 `/etc/system` 文件中删除所有的 VM 系统调优。请使用缺省设置运行，并确定是否有必要调整这些参数。请勿设置 `cachefree` 或 `priority_paging`。从 Solaris 9 发行版开始，已经删除了这两个参数。

从 Solaris 7 5/99 发行版开始，支持 CPU 和内存的动态重新配置 (dynamic reconfiguration, DR)。执行涉及添加或删除内存的 DR 操作的系统将重新计算相关参数的值，除非已在 `/etc/system` 中显式设置参数。如果已显式设置，将使用 `/etc/system` 中指定的值，除非违反了有关变量值的约束。如果违反了有关约束，将重置值。

lotsfree

说明	用作开始系统分页的初始触发器。超过此阈值时，会唤醒页面扫描程序以开始查找要回收的内存页。
数据类型	无符号长整数
缺省值	物理内存的 1/64 与 512 KB 之间的较大值
范围	最小值为 512 KB 与物理内存的 1/64 之间的较大值，以页表示，其页大小为 `getpagesize` 的返回值。有关更多信息，请参见 `getpagesize`(3C)。 最大值为物理内存页数。最大值不应大于物理内存的 30%。系统不会强制执行此范围，“验证”部分介绍的内容除外。
单位	页
是否为动态	是，但是如果执行基于内存的 DR 操作，则动态更改会丢失。
验证	如果 `lotsfree` 大于物理内存量，则会将值重置为缺省值。
含义	应始终维护以下关系：`lotsfree` 大于 `desfree`，而后者大于 `minfree`。
何时更改	页需求急剧增加，而内存算法可能无法跟上需求时。一种解决方法是提前一段时间开始回收内存。此解决方案为分页系统提供了额外的时间。 一种单凭经验的方法是将此参数设置为系统需要在几秒时间内分配的内存页数的 2 倍。此参数与工作负荷相关。DBMS 服务器使用缺省设置可以很好地运行。但是，对于执行超负荷文件系统 I/O 的系统，可能需要调整此参数。 对于工作负荷相对稳定以及内存量较大的系统，可以降低此值。可接受的最小值为 512 KB，以页表示，其页大小为 `getpagesize` 的返回值。
承诺级别	不稳定

desfree

说明　指定系统上始终可用的首选内存量。

数据类型　无符号整数

缺省值　lotsfree / 2

范围　最小值为 256 KB 与物理内存的 1/128 之间的较大值，以页表示，其页大小为 getpagesize 的返回值。

最大值为物理内存页数。最大值不应大于物理内存的 15%。系统不会强制执行此范围，“验证”部分介绍的内容除外。

单位　页

是否为动态　是，除非执行添加或删除内存的动态重新配置操作。此时，将值重置为 /etc/system 文件中提供的值或者根据新物理内存值计算得出的值。

验证　如果 desfree 大于 lotsfree，则将 desfree 设置为 lotsfree / 2。不显示任何消息。

含义　应始终维护以下关系：lotsfree 大于 desfree，而后者大于 minfree。

负面影响　增大此参数的值会产生多种负面影响。当新值接近或超过系统上的可用内存量时，将出现以下情况：

- 无法处理异步 I/O 请求，除非可用内存超过 desfree。增大 desfree 的值可能会导致拒绝原本可以成功执行的请求。
- 将 NFS 异步写入作为同步写入执行。
- 较早地唤醒交换程序，并且交换程序的行为偏向于更加主动的操作。
- 系统可能无法预先测出许多可执行页进入系统时出现的故障。这种负面影响会导致应用程序的运行速度可能慢于原本可以达到的速度。

何时更改　对于工作负荷相对稳定以及内存量较大的系统，可以降低此值。可接受的最小值为 256 KB，以页表示，其页大小为 getpagesize 的返回值。

承诺级别　不稳定

minfree

说明　指定可接受的最低内存级别。当内存低于此数值时，系统会偏向于执行以下两种分配：成功完成页出操作所需的分配或将进程完全交换出内存所需的分配。这两种分配都会拒绝或阻止其他分配请求。

数据类型　无符号整数

缺省值　desfree / 2

范围	最小值为 128 KB 与物理内存的 1/256 之间的较大值，以页表示，其页大小为 getpagesize 的返回值。 最大值为物理内存页数。最大值不应大于物理内存的 7.5%。系统不会强制执行此范围，“验证”部分介绍的内容除外。
单位	页
是否为动态	是，除非执行添加或删除内存的动态重新配置操作。此时，将值重置为 /etc/system 文件中提供的值或者根据新物理内存值计算得出的值。
验证	如果 minfree 大于 desfree，则将 minfree 设置为 desfree / 2。不显示任何消息。
含义	应始终维护以下关系：lotsfree 大于 desfree，而后者大于 minfree。
何时更改	缺省值通常已经足够。对于工作负荷相对稳定以及内存量较大的系统，可以降低此值。可接受的最小值为 128 KB，以页表示，其页大小为 getpagesize 的返回值。
承诺级别	不稳定

throttlefree

说明	指定内存级别，当达到此内存级别时，会将阻止内存分配请求置于休眠状态，即使内存足以满足此类请求时也是如此。
数据类型	无符号整数
缺省值	minfree
范围	最小值为 128 KB 与物理内存的 1/256 之间的较大值，以页表示，其页大小为 getpagesize 的返回值。 最大值为物理内存页数。最大值不应大于物理内存的 4%。系统不会强制执行此范围，“验证”部分介绍的内容除外。
单位	页
是否为动态	是，除非执行添加或删除内存的动态重新配置操作。此时，将值重置为 /etc/system 文件中提供的值或者根据新物理内存值计算得出的值。
验证	如果 throttlefree 大于 desfree，则将 throttlefree 设置为 minfree。不显示任何消息。
含义	应始终维护以下关系：lotsfree 大于 desfree，而后者大于 minfree。
何时更改	缺省值通常已经足够。对于工作负荷相对稳定以及内存量较大的系统，可以降低此值。可接受的最小值为 128 KB，以页表示，其页大小为 getpagesize 的返回值。有关更多信息，请参见 getpagesize(3C)。
承诺级别	不稳定

pageout_reserve

说明　指定为专门使用页出线程或调度程序线程保留的页数。当可用内存小于此值时，便会针对除页出或调度程序之外的所有进程拒绝非阻止分配。页出需要具有较小的内存池供自身使用，以便可以分配执行 I/O 所需的数据结构，从而将页写入其后备存储。在 Solaris 2.6 发行版中引入了此变量，用于确保系统在内存严重不足的情况下能够执行页出操作。

数据类型　无符号整数

缺省值　throttlefree / 2

范围　最小值为 64 KB 与物理内存的 1/512 之间的较大值，以页表示，其页大小为 getpagesize(3C) 的返回值。

最大值为物理内存页数。最大值不应大于物理内存的 2%。系统不会强制执行此范围，“验证”部分介绍的内容除外。

单位　页

是否为动态　是，除非执行添加或删除内存的动态重新配置操作。此时，将值重置为 /etc/system 文件中提供的值或者根据新物理内存值计算得出的值。

验证　如果 pageout_reserve 大于 throttlefree / 2，则将 pageout_reserve 设置为 throttlefree / 2。不显示任何消息。

含义　应始终维护以下关系：lotsfree 大于 desfree，而后者大于 minfree。

何时更改　缺省值通常已经足够。对于工作负荷相对稳定以及内存量较大的系统，可以降低此值。可接受的最小值为 64 KB，以页表示，其页大小为 getpagesize 的返回值。

承诺级别　不稳定

pages_pp_maximum

说明　定义必须处于未锁定状态的页数。如果某个锁定页请求将强制可用内存低于此值，则会拒绝此请求。

数据类型　无符号长整数

缺省值　以下两者中的较大值：tune_t_minarmem + 100、引导时可用内存的 4% + 4 MB

范围　系统强制的最小值为 tune_t_minarmem + 100，但不强制最大值。

单位　页

是否为动态　是，除非执行添加或删除内存的动态重新配置操作。此时，将值重置为 /etc/system 文件中提供的值或者根据新物理内存值计算得出的值。

验证	如果 /etc/system 文件中指定的值或计算得出的缺省值小于 tune_t_minarmem + 100，则将其重置为 tune_t_minarmem + 100。 如果 /etc/system 文件中的值增大，则不会显示任何消息。仅在引导时以及执行涉及添加或删除内存的动态重新配置操作期间进行验证。
何时更改	当内存锁定请求失败或者附加到带有 SHARE_MMU 标志的共享内存段失败，但仍然看似具有足够的可用内存时。 如果值过大，则可能会导致内存锁定请求（mlock、mlockall 和 memcntl）不必要地失败。有关更多信息，请参见mlock(3C)、mlockall(3C) 和memcntl(2)。
承诺级别	不稳定
更改历史记录	有关信息，请参见第 156 页中的 “pages_pp_maximum（Solaris 9 发行版之前的 Solaris 发行版）”。

tune_t_minarmem

说明	定义为避免死锁而需要保留的最小可用驻留（而非可交换）内存。用于保留一部分内存以供 OS 的核心使用。当 OS 确定最大可用内存量时，采用此方法限制的内存将不可见。
数据类型	带符号整数
缺省值	25
范围	1 到物理内存
单位	页
是否为动态	否
验证	无。较大的值会导致浪费物理内存。
何时更改	缺省值通常已经足够。如果系统锁定并且调试信息指示没有可用的内存，可以考虑增大缺省值。
承诺级别	不稳定

fastscan

说明	定义内存压力达到最大时系统每秒查看的最大页数。
数据类型	带符号整数
缺省值	64 MB 与物理内存的 1/2 之间的较小值。
范围	1 到物理内存的一半

单位	页
是否为动态	是，除非执行添加或删除内存的动态重新配置操作。此时，将值重置为由 `/etc/system` 提供的值或者根据新物理内存值计算得出的值。
验证	最大值为 64 MB 与物理内存的 1/2 之间的较小值。
何时更改	内存不足期间（尤其是在系统处于需要大量内存的阶段或执行超负荷文件 I/O 时）首选更加主动的内存扫描时。
承诺级别	不稳定

slowscan

说明	定义尝试回收内存时系统每秒查看的最小页数。
数据类型	带符号整数
缺省值	以页为单位的物理内存的1/20 与 100 之间的较小值。
范围	1 到 `fastscan` / 2
单位	页
是否为动态	是，除非执行添加或删除内存的动态重新配置操作。此时，将值重置为 `/etc/system` 文件中提供的值或者根据新物理内存值计算得出的值。
验证	如果 `slowscan` 大于 `fastscan` / 2，则将 `slowscan` 重置为 `fastscan` / 2。不显示任何消息。
何时更改	内存不足期间（尤其是在系统处于需要大量内存的阶段）首选更加主动的内存扫描时。
承诺级别	不稳定

min_percent_cpu

说明	定义 `pageout` 可占用的最小 CPU 百分比。此参数用于确定页面扫描程序可占用的最长时间。
数据类型	带符号整数
缺省值	4
范围	1 到 80
单位	百分比
是否为动态	是
验证	无

何时更改　在具有多个 CPU 以及较大内存的系统上（这些系统处于需要大量内存的阶段）增大此值，可以使页面调度程序花更多时间来尝试查找内存。

承诺级别　不稳定

handspreadpages

说明　内存较低时，Solaris OS 使用双指针时钟算法查找候选回收页。时钟的第一个指针会扫描内存，将页标记为未使用。第二个指针距离第一个指针一段距离之后再扫描内存，检查页是否仍标记为未使用。如果是，则回收页。第一个和第二个指针之间的距离为 handspreadpages。

数据类型　无符号长整数

缺省值　fastscan

范围　1 到系统上的最大物理内存页数

单位　页

是否为动态　是。此参数要求将内核 reset_hands 参数也设置为非零值。一旦识别出 handspreadpages 的新值，便会将 reset_hands 设置为零。

验证　将值设置为物理内存量与 handspreadpages **值**之间的较小值。

何时更改　需要延长页在回收之前可能驻留的时间时。增大此值会增加两个指针之间的距离，从而会延长回收页之前的时间。

承诺级别　不稳定

pages_before_pager

说明　定义在 I/O 完成之后立即释放页（而不是存储页以备将来可能重新使用）的系统阈值部分。阈值为 lotsfree + pages_before_pager。NFS 环境也使用此阈值，以便在内存压力增加时减小其异步活动。

数据类型　带符号整数

缺省值　200

范围　1 到物理内存量

单位　页

是否为动态　否

验证　无

何时更改　针对页（这些页已实际读取或写入一次，并且未再次引用）执行完大多数 I/O 时，可能要更改此参数。将此变量设置为较大的内存量，会导致继续将页添加到可用列表。

当系统的内存压力急剧增加时，也可能要更改此参数。较大的值有助于显著缓解压力。

承诺级别　不稳定

maxpgio

说明　定义分页系统可对其进行排队的最大页 I/O 请求数。将此数值除以 4 即可得到分页系统使用的实际最大数值。此参数用于限制请求数和控制进程交换。

数据类型　带符号整数

缺省值　40

范围　1 到 1024

单位　I/0

是否为动态　否

验证　无

含义　页面调度程序发出的最大 I/O 请求数受请求缓冲区列表大小的限制，此列表大小当前为 256。

何时更改　系统的内存压力急剧增加时。如果配置了多个交换设备或者交换设备为条带化设备，则较大的值有助于更快地从压力中恢复过来。

承诺级别　不稳定

与交换相关的参数

Solaris OS 中的交换由 swapfs 伪文件系统实现。交换设备和物理内存上的空间组合被视为可用空间池，用于支持系统维护异步内存的后备存储。系统首先尝试分配磁盘设备中的空间，然后将物理内存用作后备存储。强制 swapfs 将系统内存用于后备存储时，会强制执行相应的限制以确保系统不会因为 swapfs 占用过多空间而出现死锁。

swapfs_reserve

说明　定义保留以供系统 (UID = 0) 进程使用的系统内存量。

数据类型　无符号长整数

缺省值	4 MB 与物理内存的 1/16 之间的较小值
范围	最小值为 4 MB 与物理内存的 1/16 之间的较小值，以页表示，其页大小为 `getpagesize` 的返回值。 最大值为物理内存页数。最大值不应大于物理内存的 10%。系统不会强制执行此范围，“验证”部分介绍的内容除外。
单位	页
是否为动态	否
验证	无
何时更改	通常没有必要进行更改。仅当软件提供商建议时，或者当系统进程由于无法获取交换空间而终止时进行更改。较好的解决方案是向系统中添加物理内存或其他交换设备。
承诺级别	不稳定

swapfs_minfree

说明	定义要为系统其余部分保留的可用物理内存量。当任何进程尝试保留内存将其用作交换空间，从而导致系统发现可用内存小于此容量值时，该尝试将被拒绝。采用此方法保留的页只能由内核或用户级别进程用于进行锁定分配。
数据类型	无符号长整数
缺省值	2 MB 与物理内存的 1/8 之间的较大值
范围	1 到物理内存量
单位	页
是否为动态	否
验证	无
何时更改	进程由于无法获取交换空间而失败，但系统仍具有可用内存时。
承诺级别	不稳定

内核内存分配器

Solaris 内核内存分配器可在内核内部分配内存块以供客户机使用。此分配器可创建许多大小不同的高速缓存以供其客户机使用。客户机也可以请求分配器创建专供该客户机使用的高速缓存（例如，用于分配特定大小的结构）。可以使用 `kstat -c kmem_cache` 命令来查看有关由此分配器管理的每个高速缓存的统计信息。

有时，系统可能会因内存损坏而发生混乱。内核内存分配器支持调试接口（一组标志，可对缓冲区执行各种完整性检查）。它还可收集有关分配器的信息。通过完整性检查可以比较精确地检测实际出现错误的位置。收集的信息可以为您确定系统混乱的原因提供更多的数据支持。

在系统运行期间使用标志会引起额外的系统开销并占用更多的内存。仅当怀疑出现内存损坏问题时，才应使用这些标志。

kmem_flags

说明

Solaris 内核内存分配器具有各种调试和测试选项，这些选项可在 Solaris OS 内部开发周期内广泛使用。从 Solaris 2.5 发行版开始，便可使用这些选项中的部分选项。这些选项受 kmem_flags 变量控制。可使用内核调试程序设置该变量，然后重新引导系统使其生效。Solaris 8 发行版之前的版本，由于在内核内存分配器实例化的计时以及 /etc/system 文件解析方面存在问题，因此无法在 /etc/system 文件中设置这些标志。

以下介绍了五种受支持的标志设置。

标志	设置	说明
AUDIT	0x1	分配器维护包含其最近活动历史记录的日志。记录的项数取决于是否还设置了 CONTENTS。此日志大小固定。当用完空间时，便会回收较早的记录。
TEST	0x2	分配器向释放的内存中写入模式，并在下一次分配缓冲区时检查此模式是否未更改。如果缓冲区的某部分发生更改，则内存可能由先前已分配和释放此缓冲区的客户机使用。如果确定出现覆写，则系统会发生混乱。
REDZONE	0x4	分配器在请求的缓冲区的结尾提供额外内存，并将特定模式插入此内存。释放缓冲区时，将检查此模式以查看是否在缓冲区结尾后面写入数据。如果确定出现覆写，则内核会发生混乱。
CONTENTS	0x8	释放缓冲区时，分配器最多记录 256 个字节的缓冲区内容。此标志要求还应设置 AUDIT。 从 Solaris 8 发行版开始或者在早期发行版中，通过在启动内核之前引导 kadb 并设置这些标志，/etc/system 文件可以对这些标志的数值进行逻辑相加和设置。

标志	设置	说明
LITE	0x100	分配和释放缓冲区后，执行最小完整性检查。启用时，分配器将检查是否尚未写入 redzone，是否再次释放已释放的缓冲区，以及要释放的缓冲区的大小是否为已分配的大小。从 Solaris 7 3/99 发行版开始，便提供了此标志。请不要将此标志与任何其他标志一起使用。

数据类型　带符号整数

缺省值　0（禁用）

范围　0（禁用）或 1 - 15 或 256 (0x100)

是否为动态　是。运行时进行的更改仅影响新的内核内存高速缓存。初始化系统之后，很少创建新的高速缓存。

验证　无

何时更改　怀疑发生内存损坏时

承诺级别　不稳定

常规驱动程序参数

moddebug

说明　用于显示有关装入模块过程中各个步骤的消息。

数据类型　带符号整数

缺省值　0（关闭消息）

范围　以下是最有用的值：

- 0x80000000－列显 `[un] loading...` 消息。对于每个已装入的模块，将在控制台上和 `/var/adm/messages` 文件中显示以下消息：

```
Nov 5 16:12:28 sys genunix: [ID 943528 kern.notice]

load 'sched/TS_DPTBL' id 9 loaded @ 0x10126438/

0x10438dd8 size 132/2064

Nov 5 16:12:28 sys genunix: [ID 131579 kern.notice]
```

```
installing TS_DPTBL, module id 9.
```

- 0x40000000－列显详细的错误消息。对于每个已装入的模块，将在控制台上和 /var/adm/messages 文件中显示以下消息：

```
Nov 5 16:16:50 sys krtld: [ID 284770 kern.notice]

kobj_open: can’t open /platform/SUNW,Ultra-80/kernel/

sched/TS_DPTBL

Nov 5 16:16:50 sys krtld: [ID 284770 kern.notice]

kobj_open: can’t open /platform/sun4u/kernel/sched/

TS_DPTBL

Nov 5 16:16:50 sys krtld: [ID 797908 kern.notice]

kobj_open: ’/kernel/sch...

Nov 5 16:16:50 sys krtld: [ID 605504 kern.notice]

descr = 0x2a

Nov 5 16:16:50 sys krtld: [ID 642728 kern.notice]

kobj_read_file: size=34,

Nov 5 16:16:50 sys krtld: [ID 217760 kern.notice]

offset=0

Nov 5 16:16:50 sys krtld: [ID 136382 kern.notice]

kobj_read: req 8192 bytes,

Nov 5 16:16:50 sys krtld: [ID 295989 kern.notice]

got 4224

Nov 5 16:16:50 sys krtld: [ID 426732 kern.notice]

read 1080 bytes

Nov 5 16:16:50 sys krtld: [ID 720464 kern.notice]
```

```
copying 34 bytes

Nov 5 16:16:50 sys krtld: [ID 234587 kern.notice]

count = 34

[33 lines elided]

Nov 5 16:16:50 sys genunix: [ID 943528 kern.notice]

load 'sched/TS_DPTBL' id 9 loaded @ 0x10126438/

0x10438dd8 size 132/2064

Nov 5 16:16:50 sys genunix: [ID 131579 kern.notice]

installing TS_DPTBL, module id 9.

Nov 5 16:16:50 sys genunix: [ID 324367 kern.notice]

init 'sched/TS_DPTBL' id 9 loaded @ 0x10126438/

0x10438dd8 size 132/2064
```

- 0x20000000－列显更多的详细消息。此值不会列显 `0x40000000` 标志在系统引导期间所列显信息之外的任何其他信息。但是，此值会列显有关卸载模块时释放模块的其他信息。

可以将这些值相加来设置最终值。

是否为动态　是

验证　无

何时更改　未按预期方式装入模块或在装入模块期间系统似乎挂起时。请注意，设置 `0x40000000` 之后，系统引导速度会随写入控制台的消息数的增加而大大降低。

承诺级别　不稳定

常规 I/O 参数

maxphys

说明	定义最大物理 I/O 请求大小。如果驱动程序遇到大于此大小的请求，则会将此请求分为几个大小为 maxphys 的块。文件系统可以并且确实会强加其自己的限制。
数据类型	带符号整数
缺省值	131,072 (Sun-4u) 或 57,344 (x86)。 如果 sd 驱动程序支持广泛传输，则使用值 1,048,576。ssd 驱动程序缺省情况下使用 1,048,576。
范围	特定于计算机的页大小到 MAXINT
单位	字节
是否为动态	是，但是挂载文件系统时，很多文件系统都将此值装入每挂载点数据结构。许多驱动程序都在将设备连接到特定于驱动程序的数据结构时装入此值。
验证	无
何时更改	在原始设备上执行大量 I/O 操作时（包括输入和输出）。请注意，执行 OLTP 操作的 DBMS 会导致大量的较小 I/O 操作。在这种情况下，更改 maxphys 不会提高性能。 在随时要读取或写入大量数据（大于 64 KB）的 UFS 文件系统上执行进出 I/O 操作时，也要考虑更改此参数。应优化此文件系统以改善邻接性。例如，增加柱面组的大小并减小每柱面组的 inode 数。UFS 对其传输的最大 I/O 大小强加 1 MB 的内部限制。
承诺级别	不稳定

rlim_fd_max

说明	指定对单个进程可以打开的文件描述符设置的“硬”限制。覆盖此限制需要超级用户权限。
数据类型	带符号整数
缺省值	65,536
范围	1 到 MAXINT
单位	文件描述符
是否为动态	否

验证　无

何时更改　某个进程的最大已打开文件数不足时。系统功能中的其他限制表示更多的文件描述符并不会起到可能具有的作用。例如：

- 使用标准 I/O 的 32 位程序最多可以使用 256 个文件描述符。使用标准 I/O 的 64 位程序最多可以使用 20 亿个描述符。具体而言，标准 I/O 是指 libc(3LIB) 中的 stdio(3C) 函数。
- 缺省情况下，会将 select 限制为每 fd_set 使用 1024 个描述符。有关更多信息，请参见 select(3C)。从 Solaris 7 发行版开始，可以使用较大的 fd_set 大小（小于或等于 65,536）来重新编译 32 位应用程序代码。64 位应用程序使用的 fd_set 大小为 65,536（无法更改）。

在系统范围内更改此参数的一种备用方法是使用 plimit(1) 命令。如果父进程使用 plimit 更改了其限制，则所有子进程都会继承增大的限制。此备用方法对于 inetd 之类的守护进程非常有用。

承诺级别　不稳定

更改历史记录　有关信息，请参见第 155 页中的 “rlim_fd_max（Solaris 8 发行版）”。

rlim_fd_cur

说明　定义对单个进程可以打开的文件描述符设置的“软”限制。进程可能会在运行任何 shell 期间通过使用 setrlimit() 调用或发出 limit 命令，将其文件描述符限制调整到最多为 rlim_fd_max 定义的“硬”限制的任何值。无需超级用户权限即可将此限制调整到小于或等于硬限制的任何值。

数据类型　带符号整数

缺省值　256

范围　1 到 MAXINT

单位　文件描述符

是否为动态　否

验证　与 rlim_fd_max 进行比较。如果 rlim_fd_cur 大于 rlim_fd_max，则将 rlim_fd_cur 重置为 rlim_fd_max。

何时更改　某个进程的缺省已打开文件数不足时。增大此值只表示程序可能没有必要使用 setrlimit 来增加其可用的最大文件描述符数。

承诺级别　不稳定

常规文件系统参数

ncsize

说明　定义目录名称查找高速缓存 (directory name look-up cache, DNLC) 中的项数。UFS 和 NFS 使用此参数来高速缓存已解析的路径名元素。

从 Solaris 8 6/00 发行版开始，DNLC 还可高速缓存非查询信息，这意味着它可高速缓存无法在高速缓存中找到的名称。

数据类型　带符号整数

缺省值　4 x (v.v_proc + maxusers) + 320

范围　0 到 MAXINT

单位　DNLC 项

是否为动态　否

验证　无。较大的值会导致取消挂载文件系统所花费的时间增加，因为取消挂载过程中必须刷新高速缓存中此文件系统的项。

何时更改　在 Solaris 8 6/00 发行版之前，很难确定高速缓存是否过小。可以通过记下 kstat -n ncstats 返回的项数来进行此推断。已知系统工作负荷和文件访问模式的情况下，如果此项数似乎过高，则可能是由于 DNLC 过小造成的。

从 Solaris 8 6/00 发行版开始，可以使用 kstat -n dnlcstats 命令来确定因为 DNLC 过小而从其中删除项的时间。pick_heuristic 和 pick_last 参数的总和表示因为高速缓存过小而回收的其他有效项数。

如果 ncsize 的值过大，则会直接影响系统，因为系统会基于 ncsize 的值为 DNLC 分配一组数据结构。运行 32 位内核的系统将针对 ncsize 分配 36 字节的结构，而运行 64 位内核的系统将针对 ncsize 分配 64 字节的结构。此值会进一步影响 UFS 和 NFS，除非显式设置了 ufs_ninode 和 nfs:nrnode。

承诺级别　不稳定

rstchown

说明　指示 chown 系统调用的 POSIX 语义是否有效。POSIX 语义如下：

- 进程无法更改文件的属主，除非此进程使用 UID 0 运行。
- 进程无法将文件的组拥有权更改为当前不包含此文件的组，除非此进程以 UID 0 运行。

有关更多信息，请参见 chown(2)。

数据类型	带符号整数
缺省值	1，指示使用 POSIX 语义
范围	0 = 未强制使用 POSIX 语义， 或 1 = 使用 POSIX 语义
单位	切换（开/关）
是否为动态	是
验证	无
何时更改	不需要 POSIX 语义时。请注意，关闭 POSIX 语义可能会出现各种安全漏洞。此外，还会使用户可以将文件的拥有权更改为其他用户，并且在该用户或系统管理员不介入的情况下无法检索文件。
承诺级别	过时

`dnlc_dir_enable`

说明　启用大型目录高速缓存

注 – 此参数对 NFS 文件系统没有影响。

数据类型	无符号整数
缺省值	1（启用）
范围	0（禁用）或 1（启用）
是否为动态	是，但是请勿动态更改此可调参数。如果此参数最初为禁用状态，则可以将其启用。或者，如果此参数最初为启用状态，则可以将其禁用。但是，启用，禁用，然后再启用此参数可能会导致目录高速缓存过时。
验证	否
何时更改	目录高速缓存未出现已知问题时。但是，如果出现问题，应将 `dnlc_dir_enable` 设置为 0 以禁用高速缓存。
承诺级别	不稳定

`dnlc_dir_min_size`

说明　指定一个目录可高速缓存的最少项数。

注 – 此参数对 NFS 文件系统没有影响。

数据类型	无符号整数

缺省值	40
范围	0 到 MAXUINT（无最大值）
单位	项
是否为动态	是，可以随时更改此参数。
验证	无
何时更改	如果高速缓存小型目录时出现性能问题，则应增大 `dnlc_dir_min_size`。请注意，个别文件系统对高速缓存目录可能有其自己的范围限制。例如，UFS 将目录限制为最少包含 `ufs_min_dir_cache` 字节（大约 1024 个项，假设每个项包含 16 字节）。
承诺级别	不稳定

dnlc_dir_max_size

说明　指定一个目录可高速缓存的最多项数。

注 – 此参数对 NFS 文件系统没有影响。

数据类型	无符号整数
缺省值	MAXUINT（无最大值）
范围	0 到 MAXUINT
是否为动态	是，可以随时更改此参数。
验证	无
何时更改	如果高速缓存大型目录时出现性能问题，则应增大 `dnlc_dir_max_size`。
承诺级别	不稳定

segmap_percent

说明	定义用于快速访问文件系统高速缓存的最大内存量。可用内存列表不包含此内存池。
数据类型	无符号整数
缺省值	系统启动时可用内存的 12%
范围	2 MB 到 `physmem` 的 100%
单位	物理内存的百分比
是否为动态	否

验证	无
何时更改	如果预期要执行超负荷文件系统活动，并且具有足够的可用内存，则应增大此参数的值。
承诺级别	不稳定

UFS 参数

bufhwm 和 bufhwm_pct

说明	定义用于高速缓存 I/O 缓冲区的最大内存量。这些缓冲区用于写入文件系统元数据（超级块、inode、间接块和目录）。可以根据需要分配缓冲区，直到要分配的内存量（以 KB 为单位）超过 `bufhwm` 为止。此时，将从高速缓存存储区中刷新元数据，直到回收的缓冲区足以满足请求为止。 鉴于历史原因，`bufhwm` 不需要 `ufs:` 前缀。
数据类型	带符号整数
缺省值	物理内存的 2%
范围	80 KB 到物理内存的 20% 与 2 TB 之间的较小值。因此，`bufhwm_pct` 可以介于 1 到 20 之间。
单位	`bufhwm`：KB `bufhwm_pct`：物理内存的百分比
是否为动态	否。仅在系统初始化时确定 `bufhwm` 和 `bufhwm_pct` 以计算散列桶大小。此后，将根据这些参数计算的限制（以字节为单位）存储到可在分配和解除分配缓冲区时调整此值的数据结构中。 在正在运行的系统上，不遵守封锁协议而直接尝试调整此值可能会导致错误操作。 运行时修改 `bufhwm` 或 `bufhwm_pct` 没有任何效果。

验证	如果 `bufhwm` 小于其下限 80 KB 或大于其上限（物理内存的 20%、2 TB 以及最大内核堆大小的 1/4 之中的最小者），则将其重置为上限。如果尝试使用无效值，则会在系统控制台上和 `/var/adm/messages` 文件中显示以下消息：
	
	`"binit:` *bufhwm* `(value attempted) out of range`
	
	`(`*range start..range end*`). Using` *N* `as default."`

"value attempted" 是指 _nolinebreak>/etc/system 文件中指定的值或者使用内核调试程序指定的值。*N* 是系统基于可用系统内存计算所得的值。

同样，如果 bufhwm_pct 的设置值不在 1% 到 20% 的允许范围内，则将其重置为缺省值 2%。并且，将在系统控制台上和 /var/adm/messages 文件中显示以下消息：

```
"binit: bufhwm_pct(value attempted) out of range(0..20).

       Using 2 as default."
```

如果同时将 bufhwm 和 bufhwm_pct 设置为非零值，则 bufhwm 优先。

何时更改

由于只在需要时分配缓冲区，因此，缺省设置的开销必须满足缓冲区散列头控制结构的内存分配需求。在 32 位内核上，这些结构在每个潜在缓冲区占用 52 个字节；在 64 位内核上，在每个潜在缓冲区占用 96 个字节。

在 512 MB 的 64 位内核上，散列链数计算为 10316 / 32 == 322，最多可扩展到 2 的下一个整数幂，即 512。因此，散列头将占用 512 x 96 字节，或 48 KB。散列头分配假设缓冲区大小为 32 KB。

通过使用内核调试程序查看内核中的 bfreelist 结构，可以找出缓冲池中尚未分配的内存量。结构中的关键字段为 b_bufsize，此字段是指可能的剩余内存（以字节为单位）。可以使用 mdb 命令通过 buf 宏查看此字段：

```
# mdb -kLoading modules: [ unix krtld genunix ip nfs ipc ]

> bfreelist::print "struct buf" b_bufsize

b_bufsize = 0x225800
```

在此内存为 6 GB 的系统上，bufhwm 的缺省值为 122277。不能确定使用的头结构数，因为请求的实际缓冲区大小通常大于 1 KB。但是，可以从此系统的控制结构分配中有利地回收一些空间。

512 MB 系统上的相同结构显示了 10144 KB 中只有 4 KB 尚未分配。使用 kstat -n biostats 检查 biostats kstat 时，还会发现系统具有合理的 buffer_cache_hits 与 buffer_cache_lookups 的比率。同样，系统的缺省设置也比较合理。

承诺级别

不稳定

更改历史记录

有关信息，请参见第 157 页中的 “bufhwm（Solaris 9 发行版）” 。

ndquot

说明　　定义应为 UFS 文件系统分配的配额结构数。仅当在一个或多个 UFS 文件系统上启用配额时才关系到此参数。鉴于历史原因，不需要 ufs: 前缀。

数据类型　　带符号整数

缺省值　　((maxusers x 40) / 4) + max_nprocs

范围　　0 到 MAXINT

单位　　配额结构

是否为动态　　否

验证　　无。值过大将使系统挂起。

何时更改　　缺省配额结构数不足时。在这种情况下，会在控制台上显示或在消息日志中写入以下消息：

```
dquot table full
```

承诺级别　　不稳定

ufs_ninode

说明　　指定要在内存中保存的 inode 数。Inode 将针对 UFS 进行全局高速缓存，并非以每个文件系统为基础进行高速缓存。

在这种情况下，关键参数为 ufs_ninode。此参数用于计算两个可影响 inode 高速缓存处理的关键限制。将计算高水位标记 ufs_ninode / 2 以及低水位标记 ufs_ninode / 4。

当系统使用一个 inode 运行时，可能会出现以下两种情况之一：

- inode 引用的文件不再存在于系统上，因此将删除 inode。删除 inode 之后，空间又可进行 inode 高速缓存，以供其他 inode（将从磁盘读取或针对新文件创建）使用。
- 文件仍然存在，但是不再由运行的进程引用。随后将 inode 放到空闲队列中。任何引用的页仍在内存中。

当 inode 处于空闲状态时，内核便会将空闲进程向后延迟一段时间。如果文件系统为日志文件系统，则内核还会延迟删除 inode。将有两个内核线程处理这种延迟处理，每个线程负责一个队列。

执行推迟处理时，系统会将 inode 放入删除队列或空闲队列。这两个队列由单独的线程进行处理。将 inode 放到队列时，会根据低水位标记检查队列占用率。如果队列占用率超过低水位标记，则会唤醒与队列关联的线程。唤

醒队列之后，线程便会在队列中运行，并将与 inode 关联的所有页强制迁出到磁盘从而释放 inode。当线程删除了唤醒队列时队列中 50% 的 inode 时，便会停止。

如果空闲线程无法跟上负载变化，则使用第二种机制。当系统需要查找 vnode 时，便会执行 `ufs_vget` 例程。`vget` 执行的**第一项**操作是检查空闲队列的长度。如果长度大于高水位标记，则会从空闲队列中去除两个 inode 并将它们"置于空闲状态"（刷新页并释放 inode）。`vget` 将在获取供自身使用的 inode **之前**执行此操作。

系统尝试通过以下操作进行优化：将不包含主存页的 inode 放在空闲列表的开头，将包含页的 inode 放在空闲列表的结尾。但是，系统不对列表中的其他项进行排序。将始终从空闲队列的前部删除 inode。

仅当执行同步、取消挂载或重新挂载时，才从队列中全部删除 inode。

鉴于历史原因，此参数不需要 `ufs:` 前缀。

数据类型　带符号整数

缺省值　`ncsize`

范围　0 到 MAXINT

单位　Inode

是否为动态　是

验证　如果 `ufs_ninode` 小于或等于零，则将值设置为 `ncsize`。

何时更改　当缺省 inode 数不足时。如果 `kstat -n inode_cache` 报告的 `maxsize reached` 字段的值大于 `kstat` 中的 `maxsize` 字段的值，则 `ufs_ninode` 的值可能过小。如果 inode 过于空闲，也可能会出现问题。

可以通过使用 `kstat -n inode_cache` 查看 `inode_cache kstat` 来标识过于空闲的 inode。`Thread idles` 是由后台线程置于空闲状态的 inode，而 `vget idles` 是指请求进程在使用 inode 之前就将其置于空闲状态的 inode。

承诺级别　不稳定

ufs_WRITES

说明　如果 `ufs_WRITES` 为非零值，则会检查未在文件中写入的字节数。请参见 `ufs_HW` 以确定在只有 `ufs_LW` 个字节未完成之前，应该执行写入还是延迟写入。将基于每个文件跟踪未完成的总字节数，这样如果某个文件超过限制，不会影响向其他文件中写入。

数据类型　带符号整数

缺省值　1（启用）

范围	0（禁用）或 1（启用）
单位	切换（开/关）
是否为动态	是
验证	无
何时更改	希望 UFS 写入限制完全关闭时。如果 I/O 容量不足，则禁用此参数会导致磁盘具有较长服务队列。
承诺级别	不稳定

ufs_LW 和 ufs_HW

说明	ufs_HW 指定单个文件中的未完成字节数限制值。如果未完成的字节数大于此值并且设置了 ufs_WRITES，则会延迟写入。通过根据条件变量将执行写入的线程置于休眠状态来延迟写入。 ufs_LW 是单个文件中未完成字节数的限制，如果低于此限制，则会切换其他休眠进程所依赖的条件变量。当写入完成且字节数小于 ufs_LW 时，便会切换条件变量，从而导致所有线程都在等待此变量被唤醒并尝试执行其写入。
数据类型	带符号整数
缺省值	对于 ufs_LW，为 8 x 1024 x 1024；对于 ufs_HW，为 16 x 1024 x 1024
范围	0 到 MAXINT
单位	字节
是否为动态	是
验证	无
含义	仅当 ufs_WRITES 不等于零时，ufs_LW 和 ufs_HW 才有意义。应该同时更改 ufs_HW 和 ufs_LW，以免在以下情况下进行不必要的调整：进程唤醒并且发现它们无法执行写入（当 ufs_LW 和 ufs_HW 太相近时）或者它们等待的时间可能大于所需的时间（当 ufs_LW 和 ufs_HW 相差太大时）。
何时更改	当文件系统由条带化卷组成时，请考虑更改这些值。可用的聚集带宽会很容易超过 ufs_HW 的当前值。但是，此参数并非每文件系统设置。 当 ufs_throttles 为关键数值时，也可能要考虑更改此参数。当前，ufs_throttles 只能使用内核调试程序进行访问。
承诺级别	不稳定

freebehind

说明	启用 `freebehind` 算法。启用此算法之后，如果在内存使用率过高时检测到顺序 I/O，则系统会跳过新读取的块中的文件系统高速缓存。
数据类型	布尔值
缺省值	1（启用）
范围	0（禁用）或 1（启用）
是否为动态	是
验证	无
何时更改	可以非常容易地执行 `freebehind` 算法。如果不期望执行重要的顺序文件系统活动，则禁用 `freebehind` 可确保将在文件系统页高速缓存中保留所有文件（不论规模多大）。有关更精细的调优，请参见 `smallfile`。
承诺级别	不稳定

smallfile

说明	确定文件的大小阈值，如果文件大于此值，则在 `freebehind` 算法下，不进行高速缓存保留。 大内存系统包含的内存足以高速缓存数以千计的 10 MB 文件，而不会引起严重的内存需求问题。但是，这种情况在很大程度上与应用程序相关。 `smallfile` 和 `freebehind` 参数的目标是重复使用高速缓存的信息，而不会因为高速缓存过多而导致内存不足。
数据类型	带符号整数
缺省值	32,768
范围	0 到 2,147,483,647
是否为动态	是
验证	无
何时更改	如果应用程序按顺序读取适度大小的文件并很有可能从缓冲中受益，同时系统不存在可用内存不足的压力，则增大 `smallfile`。适度大小的文件是指大小为 32 KB 到 2 GB 的文件。
承诺级别	不稳定

TMPFS 参数

tmpfs:tmpfs_maxkmem

说明	定义 TMPFS 可用于其数据结构（tmpnode 和目录项）的最大内核内存量。
数据类型	无符号长整数
缺省值	一页与物理内存的 4% 之间的较大值。
范围	一页中的字节数（对于 sun4u 系统为 8192，对于所有其他系统为 4096）到首次使用 TMPFS 时可用内核内存的 25%。
单位	字节
是否为动态	是
验证	无
何时更改	如果控制台上显示了或消息文件中写入了以下消息，则应增大此值： `tmp_memalloc: tmpfs over memory limit` TMPFS 用于其数据结构的当前内存量存储在 `tmp_kmemspace` 字段中。可以使用内核调试程序检查此字段。
承诺级别	不稳定

tmpfs:tmpfs_minfree

说明	定义 TMPFS 为系统的其余部分保留的最小交换空间量。
数据类型	带符号长整数
缺省值	256
范围	0 到最大交换空间大小
单位	页
是否为动态	是
验证	无
何时更改	要在大量使用 TMPFS 的系统上保留合理的交换空间量，可以增大此数值。当控制台或消息文件显示以下消息时，表示已达到限制： *fs-name*`: File system full, swap space limit exceeded`
承诺级别	不稳定

更改历史记录　　有关信息，请参见第 156 页中的 “tmpfs:tmpfs_minfree（Solaris 8 发行版）”。

伪终端

在 Solaris 软件中，伪终端 pty 具有以下两种用途：

- 使用 telnet、rlogin 或 rsh 命令支持远程登录
- 提供 X 窗口系统用以创建命令解释程序窗口的界面

对于桌面工作站，缺省伪终端数便已足够。因此，将主要针对可用于远程登录的 pty 数进行调优。

Solaris 的早期版本需要执行这些调优步骤，以便显式配置系统使其具有首选的 pty 数。从 Solaris 8 发行版开始，采用了新机制，从而在大多数情况下不必进行调优。现在，缺省 pty 数基于系统上的内存量。仅当限制或增加可登录到系统的用户数时，才应该更改此缺省值。

在配置过程中，将使用以下三个相关变量：

- pt_cnt－最大缺省 pty 数。
- pt_pctofmem－专用于 pty 支持结构的内核内存的百分比。如果值为零，则表示任何远程用户都无法登录到系统。
- pt_max_pty－硬性最大 pty 数。

pt_cnt 的缺省值为零，此值指示系统基于 pct_pctofmem 中指定的内存量限制登录，除非设置了 pt_max_pty。如果 pt_cnt 为非零值，则会分配 pty，直到达到此限制为止。超过此阈值时，系统便会查看 pt_max_pty。如果 pt_max_pty 具有非零值，则将其与 pt_cnt 进行比较。如果 pt_cnt 小于 pt_max_pty，则允许分配 pty。如果 pt_max_pty 为零，则将 pt_cnt 与基于 pt_pctofmem 支持的 pty 数进行比较。如果 pt_cnt 小于此值，则允许分配 pty。请注意，仅当 pt_cnt 和 ptms_ptymax 的缺省值都为零时，基于 pt_pctofmem 的限制才起作用。

要对 pty 设置不同于源自 pt_pctofmem 的最大值的硬限制，请在 /etc/system 中将 pt_cnt 和 ptms_ptymax 设置为首选 pty 数。在这种情况下，与 ptms_pctofmem 的设置无关。

要针对 pty 支持使用不同的系统内存百分比，并使操作系统可管理显式限制，请执行以下操作：

- 请勿在 /etc/system 中设置 pt_cnt 或 ptms_ptymax。
- 在 /etc/system 中将 pt_pctofmem 设置为首选百分比。例如，对于 10% 设置，设置 pt_pctofmem=10。

请注意，在内存用于 pty 支持之前，实际上并不分配内存。分配内存之后，便会保持已分配状态。

pt_cnt

说明	可用 /dev/pts 项数是动态的，其上限由系统上可用物理内存量确定。pt_cnt 是用于确定系统可容纳的最小登录数的三个变量之一。系统可支持的最大缺省 /dev/pts 设备数是在引导时通过计算适合系统内存某百分比的 pty 结构数来确定的（请参见 pt_pctofmem）。如果 pt_cnt 为零，则系统分配的最大内存为此最大值。如果 pt_cnt 为非零值，则系统分配的最大内存为 pt_cnt 与缺省最大值之间的较大值。
数据类型	无符号整数
缺省值	0
范围	0 到 maxpid
单位	登录/窗口
是否为动态	否
验证	无
何时更改	需要显式控制可远程登录到系统的用户数时。
承诺级别	不稳定

pt_pctofmem

说明	指定为支持 /dev/pts 项数据结构可占用的最大物理内存百分比。运行 64 位内核的系统将针对每个 /dev/pts 项占用 176 字节。运行 32 位内核的系统将针对每个 /dev/pts 项占用 112 字节。
数据类型	无符号整数
缺省值	5
范围	0 到 100
单位	百分比
是否为动态	否
验证	无
何时更改	需要限制或增加可登录到系统的用户数时。如果值为零，则表示任何远程用户都无法登录到系统。
承诺级别	不稳定

pt_max_pty

说明	定义系统可提供的最大 pty 数
数据类型	无符号整数
缺省值	0（使用系统定义的最大值）
范围	0 到 MAXUINT
单位	登录/窗口
是否为动态	是
验证	无
含义	应该大于或等于 pt_cnt。当分配的 pty 数超过 pt_cnt 的值时，才检查此值。
何时更改	需要对支持的登录数设置上限（即使系统基于其当前配置值可以处理更多登录）时。
承诺级别	不稳定

STREAMS 参数

nstrpush

说明	指定可以插入（推入）STREAM 的模块数。
数据类型	带符号整数
缺省值	9
范围	9 到 16
单位	模块
是否为动态	是
验证	无
何时更改	在软件供应商的指导下更改。当 STREAM 超过其允许的推入计数时，不显示任何消息。但是会将值 EINVAL 返回到尝试执行该推入操作的程序。
承诺级别	不稳定

strmsgsz

说明	指定单个系统调用可以传递给 STREAM 的消息数据部分的最大字节数。任何超过该大小的 `write` 都会被分为多条消息。有关更多信息，请参见`write`(2)。
数据类型	带符号整数
缺省值	65,536
范围	0 到 262,144
单位	字节
是否为动态	是
验证	无
何时更改	当 `putmsg` 调用返回 `ERANGE` 时。有关更多信息，请参见 `putmsg`(2)。
承诺级别	不稳定

strctlsz

说明	指定单个系统调用可以传递给 STREAM 的消息控制部分的最大字节数。
数据类型	带符号整数
缺省值	1024
范围	0 到 MAXINT
单位	字节
是否为动态	是
验证	无
何时更改	在软件供应商的指导下更改。如果 `putmsg`(2) 调用尝试超过此限制，便会返回 `ERANGE`。
承诺级别	不稳定

System V 消息队列

System V 消息队列提供了一个消息传递接口，允许内核中创建的队列进行消息交换。Solaris 环境提供了多个接口，允许消息加入和离开队列。消息可以具有与其相关联的类型。通过加入队列操作，可以将消息列在队列的末尾，而通过离开列队操作，可以从队列删除第一条特定类型的消息，如果未指定类型，则删除第一条消息。

有关 Solaris 10 发行版中 System V 消息队列的信息，请参见第 18 页中的 “System V IPC 配置”。

有关调节这些系统资源的详细信息，请参见《System Administration Guide: Solaris Containers-Resource Management and Solaris Zones》中的第 6 章，“Resource Controls (Overview)”。

有关过时的 System V 消息队列的旧有信息，请参见第 165 页中的 “过时或已删除的参数”。

System V 信号

在 Solaris OS 中，System V 信号提供计数信号。**信号**是一个计数器，用于针对多个进程提供对共享数据对象的访问。除了标准的信号固定和释放操作，System V 信号还可以具有按需（例如，表示可用资源量）增加或减小的值。System V 信号还提供了同时对一组信号执行操作的功能，并可以使系统撤消上一个由某个已中止进程执行的操作。

有关 Solaris 10 发行版中对信号资源所做的更改的信息，请参见第 18 页中的 “System V IPC 配置”。

有关使用 Solaris 10 发行版中新增资源控制的详细信息，请参见《System Administration Guide: Solaris Containers-Resource Management and Solaris Zones》中的第 6 章，“Resource Controls (Overview)”。

有关过时的 System V 信号参数的旧有信息，请参见第 165 页中的 “过时或已删除的参数”。

System V 共享内存

使用 System V 共享内存，进程可以创建段。协作进程可以连接到内存段（受段访问权限的限制）并访问段中包含的数据。此功能通过可装入模块来实现。`/etc/system` 文件中的项必须包含 `shmsys:` 前缀。从 Solaris 7 发行版开始，`keyserv` 守护进程使用 System V 共享内存。

DBMS 供应商使用一种称为**锁定共享内存** (intimate shared memory, ISM) 的特殊类型的共享内存来实现最佳性能。当共享内存段成为 ISM 段时，便会锁定该段的内存。借助此功能，可以使用更快的 I/O 路径，并提高了内存使用率。然后在所有以 ISM 模式连接到段的进程间共享大量描述该段的内核资源。

有关 Solaris 10 发行版中对共享内存资源所做的更改的信息，请参见第 18 页中的 “System V IPC 配置”。

有关使用 Solaris 10 发行版中新增资源控制的详细信息，请参见《System Administration Guide: Solaris Containers-Resource Management and Solaris Zones》中的第 6 章，“Resource Controls (Overview)”。

有关过时的 System V 共享内存参数的旧有信息，请参见第 165 页中的 “过时或已删除的参数”。

segspt_minfree

说明	标识不能为 ISM 共享内存分配的系统内存页。
数据类型	无符号长整数
缺省值	创建首个 ISM 段时可用系统内存的 5%
范围	0 到物理内存的 50%
单位	页
是否为动态	是
验证	无。当 ISM 段占用内存时，太小的值可能导致系统挂起或性能严重下降。
何时更改	在使用 ISM 且具有大量物理内存的数据库服务器上，可以减小此参数的值。如果未使用 ISM 段，则此参数无效。在大型内存计算机上，最大值 128 MB (0x4000) 几乎肯定足够。
承诺级别	不稳定

调度

rechoose_interval

说明	指定在确定进程与其最后在上面运行的 CPU 失去所有关联之前的时钟周期数。此时间间隔过期之后，便会将任意 CPU 视为调度线程的候选 CPU。此参数只与分时类的线程相关。实时线程在第一个可用 CPU 上进行调度。
数据类型	带符号整数
缺省值	3
范围	0 到 MAXINT
是否为动态	是
验证	无
何时更改	当高速缓存很大，或者系统正在运行一个关键进程或运行一组看上去受并非由数据访问模式导致的过量高速缓存未命中的影响时。 更改此参数之前，请考虑使用从 Solaris 2.6 发行版开始提供的处理器集功能或处理器绑定。有关更多信息，请参见 `psrset`(1M) 或 `pbind`(1M)。

承诺级别	不稳定

计时器

hires_tick

说明	设置此参数之后将导致 Solaris OS 使用系统时钟频率 1000，而不是缺省值 100。
数据类型	带符号整数
缺省值	0
范围	0（禁用）或 1（启用）
是否为动态	否。导致在引导时设置新的系统计时变量。引导之后不再引用此变量。
验证	无
何时更改	希望超时的分辨率小于 10 毫秒且大于或等于 1 毫秒时。
承诺级别	不稳定

timer_max

说明	指定可用的 POSIX™ 计时器数。
数据类型	带符号整数
缺省值	32
范围	0 到 MAXINT
是否为动态	否。增大此值可能导致系统崩溃。
验证	无
何时更改	当系统提供的缺省计时器数不足时。执行 `timer_create` 系统调用时，应用程序将收到 `EAGAIN` 错误。
承诺级别	不稳定

Sun-4u 特定参数

consistent_coloring

说明

从 Solaris 2.6 发行版开始，便引入了在 UltraSPARC® (sun4u) 平台上使用不同的页放置策略的功能。页放置策略尝试通过分配物理页地址来最大程度地利用 L2 高速缓存。无论选择何种算法作为缺省算法，此算法可能生成的结果都会比其他用于特定应用程序集的算法要差。此参数可更改系统上为所有进程选定的放置算法。

根据 L2 高速缓存的大小，将内存分为多个容器。在未映射的页上第一次发生页面错误时，页放置代码便会从容器中分配一个页面。所选页取决于要使用以下三种可能的算法中的哪一种：

- 页面染色 (Page coloring)－将使用各种虚拟地址位确定从中选择页的容器。此算法为 Solaris 8 发行版中的缺省算法。需要将 consistent_coloring 设置为零才能使用此算法。对于此算法，不存在任何按进程历史记录。
- 虚拟地址=物理地址－程序中连续的页将从连续的容器中选择页。需要将 consistent_coloring 设置为 1 才能使用此算法。对于此算法，不存在任何按进程历史记录。
- 容器跳跃 (Bin-hopping)－程序中连续的页通常是每隔一个容器分配一个页面，但是此算法偶尔会跳过多个容器。需要将 consistent_coloring 设置为 2 才能使用此算法。每个进程都从随机选择的容器启动，并且保留最后所分配容器的每进程内存。

是否为动态　是

验证　无。值大于 2 会导致控制台上显示多条 WARNING: AS_2_BIN:bad consistent coloring value 消息。此后系统立即挂起。需要重新启动电源才能恢复系统。

何时更改　当系统的主工作负荷是一组长期运行的高性能计算 (high-performance computing, HPC) 应用程序时。更改此值可以优化性能。具有多个活动进程的文件服务器，数据库服务器和系统（例如，编译和分时服务器）则不会因更改而受益。

承诺级别　不稳定

tsb_alloc_hiwater_factor

说明

按以下方式初始化 tsb_alloc_hiwater，以便设置可分配给转换存储缓冲区 (translation storage buffer, TSB) 的物理内存量的上限：

tsb_alloc_hiwater＝ 物理内存（以字节为单位）/tsb_alloc_hiwater_factor

当分配给 TSB 的内存与 tsb_alloc_hiwater 的值相等时，由于页面未映射，因此 TSB 内存分配算法将尝试回收 TSB 内存。

使用此因子增大 tsb_alloc_hiwater 值时务必谨慎。要防止系统挂起，生成的上限值必须远低于 swapfs_minfree 和 segspt_minfree 的值。

数据类型　整数

缺省值　32

范围　1 到 MAXINT

请注意，如果因子为 1，则所有物理内存均可分配给 TSB，这可能导致系统挂起。如果因子太大，则没有可分配给 TSB 的内存，从而降低了系统性能。

是否为动态　是

验证　无

何时更改　如果系统具有许多关联到非常大的共享内存段的进程，则更改此参数的值。在大多数情况下，无需调节此变量。

承诺级别　不稳定

default_tsb_size

说明　选择分配给所有进程的初始转换存储缓冲区 (translation storage buffer, TSB) 的大小。

数据类型　整数

缺省值　缺省值为 0 (8 KB)

范围　可能的值为：

值	说明
0	8 KB
1	16 KB
3	32 KB
4	128 KB
5	256 KB

值	说明
6	512 KB
7	1 MB

是否为动态　是

验证　无

何时更改　该值通常不需要更改。但是，如果系统上多数进程的工作集都大于平均工作集，或者禁用了驻留集大小 (resident set size, RSS) 大小调整功能，则更改此值会很有益。

承诺级别　不稳定

enable_tsb_rss_sizing

说明　启用基于驻留集大小 (resident set size, RSS) 的 TSB 大小调整试探性算法。

数据类型　布尔值

缺省值　1（可以调整 TSB 的大小）

范围　0（可以调整 TSB 的大小）或 1（TSB 保留为 `tsb_default_size`）

是否为动态　是

验证　是

何时更改　请勿修改此可调参数。

承诺级别　不稳定

tsb_rss_factor

说明　控制 RSS 大小调整试探性算法的 RSS 到 TSB 的跨度比。此因子除以 512 便可得出 TSB 跨度的百分比，在将 TSB 视为备选的大小调整对象之前，此跨度比必须驻留在内存中。

数据类型　整数

缺省值　384，由于预期某些虚拟地址会映射到 TSB 中的同一个槽，因此只能得出值 75%。

范围　0 到 512

是否为动态　是

验证　无

何时更改　如果具有较小地址空间的应用程序发现由于 TSB 中的虚拟地址冲突而导致的 TSB 未命中，则可能需要考虑将此值减小至 0。

例如，在某些情况下，将 `tsb_rss_factor` 更改为 50% 而不是 75% 可能有助于消除 TSB 中的虚拟地址冲突，但是会使用更多的内核内存，特别是在高负荷的系统上。

承诺级别　不稳定

Solaris Volume Manager 参数

md_mirror:md_resync_bufsz

说明　在缓冲区中将用于重新同步 RAID 1 卷（镜像）的缓冲区大小设置为以 512 字节为单位的块数。设置更大的值可以加快重新同步的速度。

数据类型　整数

缺省值　缺省值为 128，对于小型系统尚可接受。较大的系统可能使用更大的值来加快镜像重新同步的速度。

范围　128 到 2048

单位　块（以 512 字节为单位）

是否为动态　否

验证　无

何时更改　使用 Solaris Volume Manager RAID 1 卷（镜像），并且希望加快镜像重新同步的速度时。假设您所具有的内存足以满足总体系统性能，则可以增大此值而不会导致其他性能问题。

如果需要加快镜像重新同步的速度，请以增量方式增大此参数的值（使用增量 128 块），直到获得满意的性能。在相当大的系统或新系统上，值 2048 可能为最佳值。在较旧的系统上设置较高的值可能会导致系统挂起。

承诺级别　不稳定

md:mirrored_root_flag

说明　如果有任何有效状态数据库副本可用，则将覆盖 Solaris Volume Manager 要求的副本定额并强制启动 Solaris Volume Manager。

缺省值为禁用，即要求启动 Solaris Volume Manager 之前大多数副本可用且已同步。

数据类型　布尔值

缺省值　0（禁用）

范围　0（禁用）或 1（启用）

是否为动态　否

验证　无

何时更改　不支持此参数的用法。

如果符合以下所有三个条件，则使用 Solaris Volume Manager 的用户可接受启用此参数的风险：

- 已镜像根 (/) 文件系统或其他对系统非常关键的文件系统
- 只有两个磁盘或控制器可用
- 需要执行无人参与型系统重新引导

如果启用此参数，则系统可能会使用不能正确表示系统状态（包括哪些镜像端完好，或哪些镜像端处于维护状态）的过时副本进行引导。这种情况可能会导致数据损坏或系统崩溃。

仅当系统可用性比数据一致性和完整性更重要时，才应更改此参数。密切监视系统中的任何故障。可以通过尽可能减少故障卷的个数、维护卷的个数或热交换卷的个数来减小风险。

有关状态数据库副本的更多信息，请参见《Solaris Volume Manager Administration Guide》中的第 6 章，"State Database (Overview)"。

承诺级别　不稳定

网络驱动程序参数

intr_blank_time 和 intr_blank_packets

说明　在 SPARC 系统上，这些参数影响板上网络吞吐量和等待时间。

如果禁用了中断抑制 (interrupt blanking) 功能，则包会在到达目的地时立即由驱动程序进行处理，这样会提高网络吞吐量并缩短等待时间，但是需要占用更多的 CPU。在禁用中断抑制 (interrupt blanking) 功能的情况下，在某些高负荷的 Web 服务器环境中，处理器使用率可以高达 80%–90%。

如果启用中断抑制 (interrupt blanking) 功能，则包会在发出中断命令时进行处理。启用中断抑制 (interrupt blanking) 功能会降低处理器使用率和网络吞吐量，并延长网络等待时间。

应该同时设置这两个参数。可使用 ndd 命令按如下方式设置这些参数：

```
# ndd -set /dev/eri intr_blank_time 0
```

```
# ndd -set /dev/eri intr_blank_packets 0
```

可按如下方式将它们添加到 /etc/system 文件中：

```
set eri:intr_blank_time 0
```

```
set eri:intr_blank_packets 0
```

缺省值　　在具有 eri 驱动程序的 SPARC 系统上，这两个参数为启用状态。

在具有 hme 驱动程序的 SPARC 系统上，这两个参数为禁用状态。

范围　　0（禁用）或 1（启用）

是否为动态　　是

验证　　无

何时更改　　中断抑制 (interrupt blanking) 参数值综合考虑了网络吞吐量和处理器使用率。如果为了获取较高的网络吞吐量而可以接受较高的处理器使用率，则可禁用中断抑制 (interrupt blanking) 功能。如果首选较低的处理器使用率并且可以接受网络等待时间延长，则可启用中断抑制 (interrupt blanking) 功能。

承诺级别　　不稳定

第 3 章

NFS 可调参数

本章介绍 NFS 可调参数。

有关可调参数的参考信息

可调参数	参考
Solaris 内核可调参数	第 2 章
Internet 协议套件可调参数	第 4 章
网络高速缓存和加速器 (Network Cache and Accelerator, NCA) 可调参数	第 5 章

调优 NFS 环境

可以在 /etc/system 文件（在引导过程中读取）中定义 NFS 参数。每个参数都包含其关联的内核模块的名称。有关更多信息，请参见第 22 页中的 “调优 Solaris 系统”。



注意 – 在不同的发行版中，参数名称、参数所在的模块以及缺省值可能会有所不同。更改或应用先前版本中的值之前，请查看文档以了解现用 SunOS 发行版的版本。

NFS 模块参数

本节介绍与 NFS 内核模块相关的参数。

nfs:nfs3_pathconf_disable_cache

说明	控制对已挂载 NFS 版本 3 的文件系统的 pathconf 信息的高速缓存。
数据类型	整数（32 位）
缺省值	0（启用高速缓存）
范围	0（启用高速缓存）或 1（禁用高速缓存）
单位	布尔值
是否为动态	是
验证	无
何时更改	基于每个文件高速缓存 pathconf 信息。但是，如果服务器可以动态更改特定文件的信息，应使用此参数禁用高速缓存。客户机没有用于验证其高速缓存项的机制。
承诺级别	不稳定

nfs:nfs4_pathconf_disable_cache

说明	控制对已挂载 NFS 版本 4 的文件系统的 pathconf 信息的高速缓存。
数据类型	整数（32 位）
缺省值	0（启用高速缓存）
范围	0（启用高速缓存）或 1（禁用高速缓存）
单位	布尔值
是否为动态	是
验证	无
何时更改	基于每个文件高速缓存 pathconf 信息。但是，如果服务器可以动态更改特定文件的信息，应使用此参数禁用高速缓存。客户机没有用于验证其高速缓存项的机制。
承诺级别	不稳定

nfs:nfs_allow_preepoch_time

说明　控制具有错误时间标记或**负**时间标记的文件在客户机上是否可见。

以前，NFS 客户机或 NFS 服务器都不对返回的文件时间执行任何范围检查。线上时间标记值不带符号并且长度为 32 位。因此，所有值都合法。

但是，在运行 32 位 Solaris 内核的系统上，时间标记值带有符号并且长度为 32 位。因此，可以存在表示 1970 年 1 月 1 日之前或**纪元之前**的时间标记。

在运行 64 位 Solaris 内核的系统上，问题稍有不同。64 位 Solaris 内核上的时间标记值带有符号并且长度为 64 位。无法确定时间字段表示的是完整的 32 位时间还是负时间（即 1970 年 1 月 1 日之前的时间）。

从 32 位转换为 64 位时，无法确定是否要对时间值进行符号扩展。如果时间值确实为负数，则应该对此时间值进行符号扩展。但是，如果时间值实际上表示完整的 32 位时间值，则不应对此时间值进行符号扩展。只需禁用完整的 32 位时间值即可解决此问题。

数据类型　整数（32 位）

缺省值　0（禁用 32 位时间标记）

范围　0（禁用 32 位时间标记）或 1（启用 32 位时间标记）

单位　布尔值

是否为动态　是

验证　无

何时更改　即使在正常运行期间，也可以将某些文件的时间标记值设置为将来很长的时间或过去很长的时间。如果希望使用已挂载 NFS 的文件系统访问这些文件，应将此参数设置为 1 以允许不经过检查便可传递时间标记值。

承诺级别　不稳定

nfs:nfs_cots_timeo

说明　控制已挂载 NFS 版本 2 的文件系统的缺省 RPC 超时时间，此文件系统使用面向连接的传输协议（如 TCP）。

数据类型　带符号整数（32 位）

缺省值　600（60 秒）

范围　0 到 2^{31} - 1

单位　十分之一秒

是否为动态	是，但文件系统的 RPC 超时是在挂载文件系统时设置的。要影响特定文件系统，请在更改此参数之后取消挂载然后再挂载该文件系统。
验证	无
何时更改	TCP 很好地确保了请求和响应的传输能够顺利进行。但是，如果在非常慢的网络中往返时间很长，则 NFS 版本 2 客户机可能在传输完成之前超时。 增大此参数可以防止客户机出现超时错误。此值的范围很大，因此，若此值的增幅过大，可能会导致很长一段时间内无法检测到重新传输。
承诺级别	不稳定

nfs:nfs3_cots_timeo

说明	控制已挂载 NFS 版本 3 的文件系统的缺省 RPC 超时时间，此文件系统使用面向连接的传输协议（如 TCP）。
数据类型	带符号整数（32 位）
缺省值	600（60 秒）
范围	0 到 2^{31} - 1
单位	十分之一秒
是否为动态	是，但文件系统的 RPC 超时是在挂载文件系统时设置的。要影响特定文件系统，请在更改此参数之后取消挂载然后再挂载该文件系统。
验证	无
何时更改	TCP 很好地确保了请求和响应的传输能够顺利进行。但是，如果在非常慢的网络中往返时间很长，则 NFS 版本 3 客户机可能在传输完成之前超时。 增大此参数可以防止客户机出现超时错误。此值的范围很大，因此，若此值的增幅过大，可能会导致很长一段时间内无法检测到重新传输。
承诺级别	不稳定

nfs:nfs4_cots_timeo

说明	控制已挂载 NFS 版本 4 的文件系统的缺省 RPC 超时时间，此文件系统使用面向连接的传输协议（如 TCP）。 NFS 版本 4 协议规范禁止通过同一 TCP 连接进行重新传输。因此，此参数主要控制客户机响应某些事件的速度，如检测强制执行的取消挂载操作，或检测服务器故障转移到新服务器的速度。
数据类型	带符号整数（32 位）

缺省值	600（60 秒）
范围	0 到 2^{31} - 1
单位	十分之一秒
是否为动态	是，但此参数是在挂载文件系统时设置的。要影响特定文件系统，请在更改此参数之后取消挂载然后再挂载该文件系统。
验证	无
何时更改	TCP 很好地确保了请求和响应的传输能够顺利进行。但是，如果在非常慢的网络中往返时间很长，则 NFS 版本 4 客户机可能在传输完成之前超时。 增大此参数可以防止客户机出现超时错误。此值的范围很大，因此，若此值的增幅过大，可能会导致很长一段时间内无法检测到重新传输。
承诺级别	不稳定

nfs:nfs_do_symlink_cache

说明	控制是否针对已挂载 NFS 版本 2 的文件系统高速缓存了符号链接文件的内容。
数据类型	整数（32 位）
缺省值	1（启用高速缓存）
范围	0（禁用高速缓存）或 1（启用高速缓存）
单位	布尔值
是否为动态	是
验证	无
何时更改	如果服务器更改了符号链接文件的内容但未更新文件的修改时间标记，或者时间标记的粒度过大，则客户机可能在很长一段时间内无法看到对符号链接文件的内容所做的更改。在这种情况下，应使用此参数来禁用符号链接内容的高速缓存。这样，运行在客户机上的应用程序便可以立即看到所做的更改。
承诺级别	不稳定

nfs:nfs3_do_symlink_cache

说明	控制是否针对已挂载 NFS 版本 3 的文件系统高速缓存了符号链接文件的内容。
数据类型	整数（32 位）

缺省值	1（启用高速缓存）
范围	0（禁用高速缓存）或 1（启用高速缓存）
单位	布尔值
是否为动态	是
验证	无
何时更改	如果服务器更改了符号链接文件的内容但未更新文件的修改时间标记，或者时间标记的粒度过大，则客户机可能在很长一段时间内无法看到对符号链接文件的内容所做的更改。在这种情况下，应使用此参数来禁用符号链接内容的高速缓存。这样，运行在客户机上的应用程序便可以立即看到所做的更改。
承诺级别	不稳定

nfs:nfs4_do_symlink_cache

说明	控制是否针对已挂载 NFS 版本 4 的文件系统高速缓存了符号链接文件的内容。
数据类型	整数（32 位）
缺省值	1（启用高速缓存）
范围	0（禁用高速缓存）或 1（启用高速缓存）
单位	布尔值
是否为动态	是
验证	无
何时更改	如果服务器更改了符号链接文件的内容但未更新文件的修改时间标记，或者时间标记的粒度过大，则客户机可能在很长一段时间内无法看到对符号链接文件的内容所做的更改。在这种情况下，应使用此参数来禁用符号链接内容的高速缓存。这样，运行在客户机上的应用程序便可以立即看到所做的更改。
承诺级别	不稳定

nfs:nfs_dynamic

说明	控制是否针对使用无连接传输（如 UDP）的已挂载 NFS 版本 2 的文件系统启用称为**动态重新传输**的功能。此功能尝试通过监视服务器响应时间并相应调整 RPC 超时和读写传输大小来减少重新传输次数。
数据类型	整数（32 位）

缺省值 1（启用）

范围 0（禁用）或 1（启用）

是否为动态 是，但此参数是在挂载每个文件系统时设置的。要影响特定文件系统，请在更改此参数之后取消挂载然后再挂载该文件系统。

验证 无

何时更改 请勿更改此参数。

承诺级别 不稳定

nfs:nfs3_dynamic

说明 控制是否针对使用无连接传输（如 UDP）的已挂载 NFS 版本 3 的文件系统启用称为**动态重新传输**的功能。此功能尝试通过监视服务器响应时间并相应调整 RPC 超时和读写传输大小来减少重新传输次数。

数据类型 整数（32 位）

缺省值 0（禁用）

范围 0（禁用）或 1（启用）

单位 布尔值

是否为动态 是，但此参数是在挂载每个文件系统时设置的。要影响特定文件系统，请在更改此参数之后取消挂载然后再挂载该文件系统。

验证 无

何时更改 请勿更改此参数。

承诺级别 不稳定

nfs:nfs4_dynamic

说明 控制是否针对使用无连接传输（如 UDP）的已挂载 NFS 版本 4 的文件系统启用称为**动态重新传输**的功能。此功能尝试通过监视服务器响应时间并相应调整 RPC 超时和读写传输大小来减少重新传输次数。

数据类型 整数（32 位）

缺省值 0（禁用）

范围 0（禁用）或 1（启用）

单位 布尔值

是否为动态	是，但此参数是在挂载每个文件系统时设置的。要影响特定文件系统，请在更改此参数之后取消挂载然后再挂载该文件系统。
验证	无
何时更改	请勿更改此参数。
承诺级别	不稳定

nfs:nfs_lookup_neg_cache

说明	控制是否针对已挂载 NFS 版本 2 的文件系统使用负的名称高速缓存 (negative name cache)。此负的名称高速缓存 (negative name cache) 记录那些进行查找时未找到的文件名。此高速缓存用于避免针对已知不存在的文件名发出通过网络查找的请求。
数据类型	整数（32 位）
缺省值	1（启用）
范围	0（禁用）或 1（启用）
单位	布尔值
是否为动态	是
验证	无
何时更改	要正确执行高速缓存，必须在使用负项之前对其进行严格检验。对于以只读方式挂载的文件系统，此一致性机制稍有放宽。假定服务器上的文件系统当前没有进行更改或更改的速度很慢，并且此类更改可以较慢地传播到客户机。在这种情况下，一致性机制便成为标准属性高速缓存机制。 如果文件系统在客户机上以只读方式挂载，但是将在服务器上发生更改，并且客户机需要立即看到这些更改，则应该使用此参数来禁用负高速缓存 (negative cache)。 如果禁用 nfs:nfs_disable_rddir_cache 参数，则通常还应该禁用此参数。有关更多信息，请参见第 97 页中的 “nfs:nfs_disable_rddir_cache”。
承诺级别	不稳定

nfs:nfs3_lookup_neg_cache

说明	控制是否针对已挂载 NFS 版本 3 的文件系统使用负的名称高速缓存 (negative name cache)。此负的名称高速缓 (negative name cache) 存记录那些进行查找时未找到的文件名。此高速缓存用于避免针对已知不存在的文件名发出通过网络查找的请求。

数据类型	整数（32 位）
缺省值	1（启用）
范围	0（禁用）或 1（启用）
单位	布尔值
是否为动态	是
验证	无
何时更改	要正确执行高速缓存，必须在使用负项之前对其进行严格检验。对于以只读方式挂载的文件系统，此一致性机制稍有放宽。假定服务器上的文件系统当前没有进行更改或更改的速度很慢，并且此类更改可以较慢地传播到客户机。在这种情况下，一致性机制便成为标准属性高速缓存机制。 如果文件系统在客户机上以只读方式挂载，但是将在服务器上发生更改，并且客户机需要立即看到这些更改，则应该使用此参数来禁用负高速缓存 (negative cache)。 如果禁用 `nfs:nfs_disable_rddir_cache` 参数，则通常还应该禁用此参数。有关更多信息，请参见第 97 页中的 “`nfs:nfs_disable_rddir_cache`” 。
承诺级别	不稳定

`nfs:nfs4_lookup_neg_cache`

说明	控制是否针对已挂载 NFS 版本 4 的文件系统使用负的名称高速缓存 (negative name cache)。此负的名称高速缓存 (negative name cache) 记录那些进行查找时未找到的文件名。此高速缓存用于避免针对已知不存在的文件名发出通过网络查找的请求。
数据类型	整数（32 位）
缺省值	1（启用）
范围	0（禁用）或 1（启用）
单位	布尔值
是否为动态	是
验证	无
何时更改	要正确执行高速缓存，必须在使用负项之前对其进行严格检验。对于以只读方式挂载的文件系统，此一致性机制稍有放宽。假定服务器上的文件系统当前没有进行更改或更改的速度很慢，并且此类更改可以较慢地传播到客户机。在这种情况下，一致性机制便成为标准属性高速缓存机制。

如果文件系统在客户机上以只读方式挂载，但是将在服务器上发生更改，并且客户机需要立即看到这些更改，则应该使用此参数来禁用负高速缓存 (negative cache)。

如果禁用 nfs:nfs_disable_rddir_cache 参数，则通常还应该禁用此参数。有关更多信息，请参见第 97 页中的 “nfs:nfs_disable_rddir_cache”。

承诺级别　不稳定

nfs:nfs_max_threads

说明　控制执行 NFS 版本 2 客户机异步 I/O 的内核线程数。由于 NFS 基于 RPC 并且 RPC 本身具有同步特性，因此，需要不同的执行上下文来执行与调用线程异步的 NFS 操作。

可以异步执行的操作包括：针对 read-ahead 请求执行的读取操作，针对 readdir read-ahead 请求执行的目录读取操作，针对 putpage 和 pageio 请求执行的写入操作，提交操作，以及针对 inactive 请求执行的清除操作。

数据类型　整数（16 位）

缺省值　8

范围　0 到 2^{15} - 1

单位　线程

是否为动态　是，但此参数是在挂载每个文件系统时设置的。要影响特定文件系统，请在更改此参数之后取消挂载然后再挂载该文件系统。

验证　无

何时更改　要增大或减小在任何指定时间未完成的同时执行的 I/O 操作数时。例如，对于带宽很小的网络，可能需要减小此值以便 NFS 客户机不会引起网络过载。或者，如果网络带宽很大，并且客户机和服务器具有足够的资源，则可能需要增大此值。这样，可以更有效地利用可用网络带宽，以及客户机和服务器的资源。

承诺级别　不稳定

nfs:nfs3_max_threads

说明　控制执行 NFS 版本 3 客户机异步 I/O 的内核线程数。由于 NFS 基于 RPC 并且 RPC 本身具有同步特性，因此，需要不同的执行上下文来执行与调用线程异步的 NFS 操作。

	可以异步执行的操作包括：针对 read-ahead 请求执行的读取操作，针对 readdir read-ahead 请求执行的目录读取操作，针对 putpage 和 pageio 请求执行的写入操作，以及提交操作。
数据类型	整数（16 位）
缺省值	8
范围	0 到 2^{15} - 1
单位	线程
是否为动态	是，但此参数是在挂载每个文件系统时设置的。要影响特定文件系统，请在更改此参数之后取消挂载然后再挂载该文件系统。
验证	无
何时更改	要增大或减小在任何指定时间未完成的同时执行的 I/O 操作数时。例如，对于带宽很小的网络，可能需要减小此值以便 NFS 客户机不会引起网络过载。或者，如果网络带宽很大，并且客户机和服务器具有足够的资源，则可能需要增大此值。这样，可以更有效地利用可用网络带宽，以及客户机和服务器的资源。
承诺级别	不稳定

nfs:nfs4_max_threads

说明	控制执行 NFS 版本 4 客户机异步 I/O 的内核线程数。由于 NFS 基于 RPC 并且 RPC 本身具有同步特性，因此，需要不同的执行上下文来执行与调用线程异步的 NFS 操作。 可以异步执行的操作包括客户机停止使用文件时针对 read-ahead 请求执行的读取操作以及后写操作、目录读前操作和清除操作。
数据类型	整数（16 位）
缺省值	8
范围	0 到 2^{15} - 1
单位	线程
是否为动态	是，但此参数是在挂载每个文件系统时设置的。要影响特定文件系统，请在更改此参数之后取消挂载然后再挂载该文件系统。
验证	无
何时更改	要增大或减小在任何指定时间未完成的同时执行的 I/O 操作数时。例如，对于带宽很小的网络，可能需要减小此值以便 NFS 客户机不会引起网络过

	载。或者，如果网络带宽很大，并且客户机和服务器具有足够的资源，则可能需要增大此值。这样，可以更有效地利用可用网络带宽，以及客户机和服务器资源。
承诺级别	不稳定

nfs:nfs_nra

说明	控制以顺序方式访问文件时，NFS 版本 2 客户机排队的读前操作数。这些读前操作可增加并行性和读取吞吐量。每个读前请求通常都对应于一个文件数据逻辑块。
数据类型	整数（32 位）
缺省值	4
范围	0 到 2^{31} - 1
单位	逻辑块。（请参见第 98 页中的“nfs:nfs_bsize”。）
是否为动态	是
验证	无
何时更改	要增大或减小在任何指定时间针对特定文件未完成的读前请求数时。例如，对于带宽很小的网络或者在低内存的客户机上，可能需要减小此值以便 NFS 客户机不会引起网络过载或系统内存过载。或者，如果网络带宽很大，并且客户机和服务器具有足够的资源，则可能需要增大此值。这样，可以更有效地利用可用网络带宽，以及客户机和服务器资源。
承诺级别	不稳定

nfs:nfs3_nra

说明	控制以顺序方式访问文件时，NFS 版本 3 客户机排队的读前操作数。这些读前操作可增加并行性和读取吞吐量。每个读前请求通常都对应于一个文件数据逻辑块。
数据类型	整数（32 位）
缺省值	1
范围	0 到 2^{31} - 1
单位	逻辑块。（请参见第 98 页中的“nfs:nfs3_bsize”。）
是否为动态	是
验证	无

何时更改　要增大或减小在任何指定时间针对特定文件未完成的读前请求数时。例如，对于带宽很小的网络或者在低内存的客户机上，可能需要减小此值以便 NFS 客户机不会引起网络过载或系统内存过载。或者，如果网络带宽很大，并且客户机和服务器具有足够的资源，则可能需要增大此值。这样，可以更有效地利用可用网络带宽，以及客户机和服务器资源。

承诺级别　不稳定

nfs:nfs4_nra

说明　控制以顺序方式访问文件时，NFS 版本 4 客户机排队的读前操作数。这些读前操作可增加并行性和读取吞吐量。每个读前请求通常都对应于一个文件数据逻辑块。

数据类型　整数（32 位）

缺省值　4

范围　0 到 2^{31} - 1

单位　逻辑块。（请参见第 99 页中的 “nfs:nfs4_bsize” 。）

是否为动态　是

验证　无

何时更改　要增大或减小在任何指定时间针对特定文件未完成的读前请求数时。例如，对于带宽很小的网络或者在低内存的客户机上，可能需要减小此值以便 NFS 客户机不会引起网络过载或系统内存过载。或者，如果网络带宽很大，并且客户机和服务器具有足够的资源，则可能需要增大此值。这样，可以更有效地利用可用网络带宽，以及客户机和服务器的资源。

承诺级别　不稳定

nfs:nrnode

说明　控制 NFS 客户机上 rnode 高速缓存的大小。

NFS 版本 2、3 和 4 客户机使用的 rnode 是用于描述 NFS 客户机上文件的中心数据结构。rnode 包含用于标识服务器上的文件的文件句柄。rnode 还包含指向 NFS 客户机所使用的各种高速缓存的指针，以避免对服务器进行网络调用。rnode 与 vnode 一对一关联。vnode 高速缓存文件数据。

NFS 客户机尝试维护最小 rnode 数，以尝试避免销毁高速缓存的数据和元数据。重用或释放 rnode 时，一定会销毁高速缓存的数据和元数据。

数据类型　整数（32 位）

缺省值	此参数的缺省设置为 0，这意味着应该将 nrnode 的值设置为 ncsize 参数的值。实际上，nrnode 的任何非正值都会导致将 nrnode 设置为 ncsize 的值。
范围	1 到 2^{31} - 1
单位	rnode
是否为动态	否。只能通过在 /etc/system 文件中添加或更改该参数然后重新引导系统来更改此值。
验证	系统强制使用最大值，以使 rnode 只能占用 25% 的可用内存。
何时更改	由于 rnode 的创建和销毁过程是动态的，因此系统往往会设置 *nrnode* 大小的高速缓存，从而在系统内存压力增加或者同时访问的文件数增加时自动调整高速缓存的大小。但是，在某些情况下，如果提前预测到将访问多个文件，则应该设置 nrnode 的值。例如，如果 NFS 客户机要访问几个很大的文件，则应该将 nrnode 的值设置为较小的数值，以便系统内存可以高速缓存文件数据而不是高速缓存 rnode。或者，如果客户机要访问许多较小的文件，则应该增大 nrnode 的值以优化文件元数据存储，从而减少对元数据的网络调用数。 虽然建议不要这样做，但是通过将 nrnode 的值设置为 1 可以有效地禁用 rnode 高速缓存。此值指示客户机仅高速缓存 1 个 rnode，这意味着频繁重用此 rnode。
承诺级别	不稳定
更改历史记录	有关信息，请参见第 158 页中的 “nfs:nrnode (Solaris 9 8/03)”。

nfs:nfs_shrinkreaddir

说明	某些早期 NFS 服务器可能会错误地处理目录信息超过 1024 字节的 NFS 版本 2 READDIR 请求。此问题是由服务器实现中存在的错误造成的。不过，此参数在 NFS 版本 2 客户机中提供了解决方法。 启用此参数时，客户机不会生成目录信息超过 1024 字节的 READDIR 请求。如果禁用此参数，则线上大小将设置为使用 getdents 系统调用传递的大小与使用 NFS_MAXDATA 传递的大小（8192 字节）之间的较小者。有关更多信息，请参见 getdents(2)。
数据类型	整数（32 位）
缺省值	0（禁用）
范围	0（禁用）或 1（启用）
单位	布尔值
是否为动态	是

验证 无

何时更改 如果仅使用早期NFS版本2服务器并且在服务器尝试读取目录时出现互操作性问题，应检查此参数的值。启用此参数可能会轻微降低读取目录的应用程序的性能。

承诺级别 不稳定

nfs:nfs3_shrinkreaddir

说明 某些早期NFS服务器可能会错误地处理目录信息超过1024字节的NFS版本3 READDIR 请求。此问题是由服务器实现中存在的错误造成的。因此，此参数包含NFS版本3客户机中的解决方法。

启用此参数时，客户机不会生成目录信息超过1024字节的 READDIR 请求。如果禁用此参数，则线上大小将设置为使用 getdents 系统调用传递的大小与使用 MAXBSIZE 传递的大小（8192字节）之间的较小者。有关更多信息，请参见 getdents(2)。

数据类型 整数（32位）

缺省值 0（禁用）

范围 0（禁用）或1（启用）

单位 布尔值

是否为动态 是

验证 无

何时更改 如果仅使用早期NFS版本3服务器并且在服务器尝试读取目录时出现互操作性问题，应检查此参数的值。启用此参数可能会轻微降低读取目录的应用程序的性能。

承诺级别 不稳定

nfs:nfs4_shrinkreaddir

说明 某些NFS服务器可能会错误处理目录信息超过1024字节的NFS版本4 READDIR 请求。此问题是由服务器实现中存在的错误造成的。因此，此参数包含NFS版本4客户机中的解决方法。

启用此参数时，客户机不会生成目录信息超过1024字节的 READDIR 请求。如果禁用此参数，则线上大小将设置为使用 getdents 系统调用传递的大小与使用 MAXBSIZE 传递的大小（8192字节）之间的较小者。有关更多信息，请参见 getdents(2)。

数据类型 整数（32位）

缺省值　0（禁用）

范围　0（禁用）或 1（启用）

单位　布尔值

是否为动态　是

验证　无

何时更改　如果仅使用 NFS 版本 4 服务器并且在服务器尝试读取目录时出现互操作性问题，应检查此参数的值。启用此参数可能会轻微降低读取目录的应用程序的性能。

承诺级别　不稳定

nfs:nfs_write_error_interval

说明　控制记录 NFS 客户机收到的 ENOSPC 和 EDQUOT 写入错误之的时间间隔。此参数会影响 NFS 版本 2、3 和 4 客户机。

数据类型　长整数（在 32 位平台上为 32 位，在 64 位平台上为 64 位）

缺省值　5 秒

范围　在 32 位平台上，为 0 到 2^{31} - 1

在 64 位平台上，为 0 到 2^{63} - 1

单位　秒

是否为动态　是

验证　无

何时更改　增大或减小此参数的值以响应客户机记录的消息量。通常，可能要增大此参数的值，以减少频繁使用服务器上空间已满的文件系统时列显的 out of space 消息数。

承诺级别　不稳定

更改历史记录　有关信息，请参见第 158 页中的 “nfs:nfs_write_error_interval (Solaris 9 8/03)”。

nfs:nfs_write_error_to_cons_only

说明　控制是将 NFS 写入错误同时记录到系统控制台和 syslog，还是只记录到系统控制台。此参数会影响 NFS 版本 2、3 和 4 客户机的消息。

数据类型　整数（32 位）

缺省值	0（系统控制台和 syslog）
范围	0（系统控制台和 syslog）或 1（系统控制台）
单位	布尔值
是否为动态	是
验证	无
何时更改	检查此参数的值以避免文件系统被 syslogd 守护进程记录的消息填满。启用此参数时，只会在系统控制台上列显消息，而不会将其复制到 syslog 消息文件。
承诺级别	不稳定
更改历史记录	有关信息，请参见第 158 页中的 "nfs:nfs_write_error_to_cons_only (Solaris 9 8/03)"。

nfs:nfs_disable_rddir_cache

说明	控制是否使用保存来自 READDIR 和 READDIRPLUS 请求的响应的高速缓存。此高速缓存可避免对服务器进行线上调用以检索目录信息。
数据类型	整数（32 位）
缺省值	0（启用高速缓存）
范围	0（启用高速缓存）或 1（禁用高速缓存）
单位	布尔值
是否为动态	是
验证	无
何时更改	如果在某个目录中创建文件或目录时，或者删除某个目录中的文件或目录时服务器未更新该目录的修改时间，从而出现互操作性问题，则检查此参数的值。症状包括已添加到目录的新名称未出现在目录列表中，或者已从目录中删除的旧名称仍然出现在目录列表中。 此参数控制已挂载 NFS 版本 2、3 和 4 的文件系统的高速缓存。此参数应用于所有已挂载 NFS 的文件系统，因此，不能基于每个文件系统禁用或启用高速缓存。 如果禁用此参数，则还应该禁用以下参数以防止 DNLC 负高速缓存 (negative cache) 中出现坏项： ■ 第 88 页中的 "nfs:nfs_lookup_neg_cache" ■ 第 88 页中的 "nfs:nfs3_lookup_neg_cache" ■ 第 89 页中的 "nfs:nfs4_lookup_neg_cache"

承诺级别	不稳定
更改历史记录	有关信息，请参见第 159 页中的“nfs:nfs_disable_rddir_cache (Solaris 9 8/03)”。

nfs:nfs_bsize

说明	控制 NFS 版本 2 客户机使用的逻辑块大小。此块大小表示客户机需要执行 I/O 操作时尝试从服务器读取或写入服务器的数据量。
数据类型	无符号整数（32 位）
缺省值	8192 字节
范围	0 到 2^{31} - 1
单位	字节
是否为动态	是，但文件系统的块大小是在挂载文件系统时设置的。要影响特定文件系统，请在更改此参数之后取消挂载然后再挂载该文件系统。
验证	无。将此参数设置过低或过高都可能会导致系统无法正常运行。请勿将此参数设置为小于特定平台的 PAGESIZE 的任何值。请勿将此参数设置过高，因为这样可能会导致系统在等待授予内存分配时挂起。
何时更改	请勿更改此参数。
承诺级别	不稳定

nfs:nfs3_bsize

说明	控制 NFS 版本 3 客户机使用的逻辑块大小。此块大小表示客户机需要执行 I/O 操作时尝试从服务器读取或写入服务器的数据量。
数据类型	无符号整数（32 位）
缺省值	32,768 (32 KB)
范围	0 到 2^{31} - 1
单位	字节
是否为动态	是，但文件系统的块大小是在挂载文件系统时设置的。要影响特定文件系统，请在更改此参数之后取消挂载然后再挂载该文件系统。
验证	无。将此参数设置过低或过高都可能会导致系统无法正常运行。请勿将此参数设置为小于特定平台的 PAGESIZE 的任何值。请勿将此参数设置过高，因为这样可能会导致系统在等待授予内存分配时挂起。

何时更改　尝试更改最大数据传输大小时，请检查此参数的值。将此参数与 nfs:nfs3_max_transfer_size 参数一起更改。如果首选较大的传输，则增大这两个参数。如果首选较小的传输，则只需减小此参数。

承诺级别　不稳定

nfs:nfs4_bsize

说明　控制 NFS 版本 4 客户机使用的逻辑块大小。此块大小表示客户机需要执行 I/O 操作时尝试从服务器读取或写入服务器的数据量。

数据类型　无符号整数（32 位）

缺省值　32,768 (32 KB)

范围　0 到 2^{31} - 1

单位　字节

是否为动态　是，但文件系统的块大小是在挂载文件系统时设置的。要影响特定文件系统，请在更改此参数之后取消挂载然后再挂载该文件系统。

验证　无。将此参数设置过低或过高都可能会导致系统无法正常运行。请勿将此参数设置为小于特定平台的 PAGESIZE 的任何值。请勿将此参数设置过高，因为这样可能会导致系统在等待授予内存分配时挂起。

何时更改　尝试更改最大数据传输大小时，请检查此参数的值。将此参数与 nfs:nfs4_max_transfer_size 参数一起更改。如果首选较大的传输，则增大这两个参数。如果首选较小的传输，则只需减小此参数。

承诺级别　不稳定

nfs:nfs_async_clusters

说明　控制 NFS 版本 2 客户机生成的多种异步请求。四种异步请求类型为 read-ahead、putpage、pageio 和 readdir-ahead。客户机尝试在这些不同的请求类型之间循环，以便尽量实现公平，不会为了处理一种请求而牺牲其他请求。

但是，某些 NFS 版本 2 服务器中的功能（如写入收集）取决于现有 NFS 版本 2 客户机的某些行为。具体地说，写入收集功能取决于客户机是否几乎在同时发出多个 WRITE 请求。如果每次只提取队列中的一个请求，则客户机实际上禁用了这种旨在提高客户机性能的服务器功能。

因此，可使用此参数来控制在切换类型之前应发出的每种类型的请求数。

数据类型　无符号整数（32 位）

缺省值　1

范围	0 到 2^{31} - 1
单位	异步请求
是否为动态	是，但文件系统的群集设置是在挂载文件系统时设置的。要影响特定文件系统，请在更改此参数之后取消挂载然后再挂载该文件系统。
验证	无。但是，将此参数的值设置为 0 会使得只有在处理了某一特定请求类型的所有排队请求后，才继续处理下一种类型。这实际上使算法丧失了其公平性。
何时更改	要增大在切换到下一类型之前应生成的每种类型异步请求数时。这样，可能会有助于利用取决于来自客户机的请求群集的服务器功能。
承诺级别	不稳定

nfs:nfs3_async_clusters

说明

控制 NFS 版本 3 客户机生成的多种异步请求。五种异步请求类型为 read-ahead、putpage、pageio、readdir-ahead 和 commit。客户机尝试在这些不同的请求类型之间循环，以便尽量实现公平，不会为了处理一种请求而牺牲其他请求。

但是，某些 NFS 版本 3 服务器中的功能（如写入收集）取决于现有 NFS 版本 3 客户机的某些行为。具体地说，写入收集功能取决于客户机是否几乎在同时发出多个 WRITE 请求。如果每次只提取队列中的一个请求，则客户机实际上禁用了这种旨在提高客户机性能的服务器功能。

因此，可使用此参数来控制在切换类型之前应发出的每种类型的请求数。

数据类型	无符号整数（32 位）
缺省值	1
范围	0 到 2^{31} - 1
单位	异步请求
是否为动态	是，但文件系统的群集设置是在挂载文件系统时设置的。要影响特定文件系统，请在更改此参数之后取消挂载然后再挂载该文件系统。
验证	无。但是，将此参数的值设置为 0 会使得只有在处理了某一特定请求类型的所有排队请求后，才继续处理下一种类型。此值使该算法无需顾虑是否公平的问题。
何时更改	要增大在切换到下一类型之前应生成的每种类型异步操作数时。这样，可能会有助于利用取决于来自客户机的请求群集的服务器功能。
承诺级别	不稳定

nfs:nfs4_async_clusters

说明

控制 NFS 版本 4 客户机生成的多种异步请求。六种异步请求类型为 read-ahead、putpage、pageio、readdir-ahead、commit 和 inactive。客户机尝试在这些不同的请求类型之间循环，以便尽量实现公平，不会为了处理一种请求而牺牲其他请求。

但是，某些 NFS 版本 4 服务器中的功能（如写入收集）取决于现有 NFS 版本 4 客户机的某些行为。具体地说，写入收集功能取决于几乎客户机是否在同时发出多个 WRITE 请求。如果每次只提取队列中的一个请求，则客户机实际上禁用了这种旨在提高客户机性能的服务器功能。

因此，可使用此参数来控制在切换类型之前应发出的每种类型的请求数。

数据类型

无符号整数（32 位）

缺省值

1

范围

0 到 2^{31} - 1

单位

异步请求

是否为动态

是，但文件系统的群集设置是在挂载文件系统时设置的。要影响特定文件系统，请在更改此参数之后取消挂载然后再挂载该文件系统。

验证

无。但是，将此参数的值设置为 0 会使得只有在处理了某一特定请求类型的所有排队请求后，才继续处理下一种类型。这实际上使算法丧失了其公平性。

何时更改

要增大在切换到下一类型之前生成的每种类型异步请求数时。这样，可能会有助于利用取决于来自客户机的请求群集的服务器功能。

承诺级别

不稳定

nfs:nfs_async_timeout

说明

控制执行异步 I/O 请求的线程在退出之前未执行任何操作而保持休眠状态的持续时间。如果没有更多的请求可以执行，则每个线程便会进入休眠状态。如果在此计时器到期之前没有新的请求进入，则线程便会被唤醒并退出。如果某一请求到达，则会唤醒线程以执行请求，直到再次没有可执行的请求为止。然后，线程再次进入休眠状态以等待其他请求到达，或者等待计时器到期。

数据类型

整数（32 位）

缺省值

6000（时间为 1 分钟，表示为 60 秒 * 100 Hz）

范围

0 到 2^{31} - 1

单位

Hz。（通常，时钟的运行频率为 100 Hz。）

是否为动态　是

验证　无。但是，将此参数设置为非正值会导致这些线程在队列中没有可处理的请求时立即退出。

何时更改　如果确切了解系统中应用程序的行为并且可以预测异步 I/O 请求的频率，则可以采用以下任一种方法调优此参数来稍微优化性能：

- 使线程尽快到期，从而尽快释放内核资源
- 使线程到期的速度放慢，从而避免创建和销毁线程时产生的开销

承诺级别　不稳定

nfs:nacache

说明　调整访问 NFS 客户机上文件访问高速缓存的散列队列数。文件访问高速缓存是存储用户对其尝试访问的文件拥有的文件访问权限。该高速缓存自身是动态分配的。但是，用于建立指向高速缓存的索引的散列队列则是静态分配的。该算法假设每个活动文件具有一个访问高速缓存项，每个散列桶具有四个这样的访问高速缓存项。因此，缺省情况下，此参数的值被设置为 nrnode 参数的值。

数据类型　整数（32 位）

缺省值　此参数的缺省设置为 0。此值意味着应该将 nacache 的值设置为 nrnode 参数的值。

范围　1 到 2^{31} - 1

单位　访问高速缓存项

是否为动态　否。只能通过在 /etc/system 文件中添加或更改此参数然后重新引导系统来更改此值。

验证　无。但是，将此参数设置为负值可能会导致系统尝试分配一个很大的散列队列组。尝试执行此操作时，系统可能会挂起。

何时更改　如果违反了每个文件具有一个访问高速缓存项这样的基本假设，应检查此参数的值。系统处于多个用户几乎同时访问同一文件的分时模式时可能会出现这种违反情况。在这种情况下，增大预期的访问高速缓存大小，有助于保持散列访问高速缓存的高效率。

承诺级别　不稳定

nfs:nfs3_jukebox_delay

说明

控制 NFS 版本 3 客户机在收到先前请求的 NFS3ERR_JUKEBOX 错误之后等待传输新请求的持续时间。当文件由于某种原因而暂时不可用时，通常会从服务器返回 NFS3ERR_JUKEBOX 错误。此错误通常与分层存储、CD 或磁带自动换带机关联。

数据类型

长整数（在 32 位平台上为 32 位，在 64 位平台上为 64 位）

缺省值

1000（时间为 10 秒，表示为 10 秒 * 100 Hz）

范围

在 32 位平台上，为 0 到 2^{31} - 1

在 64 位平台上，为 0 到 2^{63} - 1

单位

Hz。（通常，时钟的运行频率为 100 Hz。）

是否为动态

是

验证

无

何时更改

检查此参数的值并可能将其调整为与服务器表现的行为相匹配。如果将文件变为可用的延迟时间很长，则应增大此值，以减少由于重复重新传输造成的网络开销。减小此值以缩短确认文件是否变为可用而引起的延迟。

承诺级别

不稳定

nfs:nfs3_max_transfer_size

说明

控制 NFS 版本 3 READ、WRITE、READDIR 或 READDIRPLUS 请求的数据部分的最大大小。此参数控制服务器返回的请求的最大大小以及客户机生成的请求的最大大小。

数据类型

整数（32 位）

缺省值

1,048,576 (1 MB)

范围

0 到 2^{31} - 1

单位

字节

是否为动态

是，但此参数是在挂载每个文件系统时设置的。要影响特定文件系统，请在更改此参数之后取消挂载然后再挂载该文件系统。

验证

无。但是，如果将服务器上的最大传输大小设置为 0，则可能会导致客户机无法正常运行，或者导致客户机决定不尝试与服务器进行对话。

使用采用 UDP 传输的 NFS，同样存在对最大传输大小的限制。UDP 针对每个数据报具有 64 KB 的硬性限制。除请求的数据部分之外，此 64 KB 还必须包括 RPC 包头以及其他 NFS 信息。将此限制设置过高可能会导致 UDP 出现错误，以及客户机与服务器之间出现通信问题。

何时更改　要调优通过网络传输的数据的大小时。通常，还应该更新 nfs:nfs3_bsize 参数以反映此参数的更改。

例如，尝试将传输大小增大到超过 32 KB 时，应更新 nfs:nfs3_bsize 以反映增大的值。否则，无法观测到对线上请求大小所做的任何更改。有关更多信息，请参见第 98 页中的“nfs:nfs3_bsize”。

如果要使用的传输大小小于缺省传输大小，请对每个文件系统使用 mount 命令的 -wsize 或 -rsize 选项。

承诺级别　不稳定

更改历史记录　有关信息，请参见第 159 页中的“nfs:nfs3_max_transfer_size (Solaris 9 8/03)”。

nfs:nfs4_max_transfer_size

说明　控制 NFS 版本 4 READ、WRITE、READDIR 或 READDIRPLUS 请求的数据部分的最大大小。此参数控制服务器返回的请求的最大大小以及客户机生成的请求的最大大小。

数据类型　整数（32 位）

缺省值　32, 768 (32 KB)

范围　0 到 $2^{31} - 1$

单位　字节

是否为动态　是，但此参数是在挂载每个文件系统时设置的。要影响特定文件系统，请在更改此参数之后取消挂载然后再挂载该文件系统。

验证　无。但是，如果将服务器上的最大传输大小设置为 0，则可能会导致客户机无法正常运行，或者导致客户机决定不尝试与服务器进行对话。

使用采用 UDP 传输的 NFS 时，同样存在对最大传输大小的限制。有关 UDP 的最大传输大小的更多信息，请参见第 103 页中的“nfs:nfs3_max_transfer_size”。

何时更改　要调优通过网络传输的数据的大小时。通常，还应该更新 nfs:nfs4_bsize 参数以反映此参数的更改。

例如，尝试将传输大小增大到超过 32 KB 时，应更新 nfs:nfs4_bsize 以反映增大的值。否则，无法观测到对线上请求大小所做的任何更改。有关更多信息，请参见第 99 页中的“nfs:nfs4_bsize”。

如果要使用的传输大小小于缺省传输大小，请对每个文件系统使用 mount 命令的 -wsize 或 -rsize 选项。

承诺级别　不稳定

nfs:nfs3_max_transfer_size_clts

说明 控制通过 UDP 传输的 NFS 版本 3 READ、WRITE、READDIR 或 READDIRPLUS 请求的数据部分的最大大小。此参数控制服务器返回的请求的最大大小以及客户机生成的请求的最大大小。

数据类型 整数（32 位）

缺省值 32,768 (32 KB)

范围 0 到 2^{31} - 1

单位 字节

是否为动态 是，但此参数是在挂载每个文件系统时设置的。要影响特定文件系统，请在更改此参数之后取消挂载然后再挂载该文件系统。

验证 无。但是，如果将服务器上的最大传输大小设置为 0，则可能会导致客户机无法正常运行，或者导致客户机决定不尝试与服务器进行对话。

何时更改 请勿更改此参数。

承诺级别 不稳定

nfs:nfs3_max_transfer_size_cots

说明 控制通过 TCP 传输的 NFS 版本 3 READ、WRITE、READDIR 或 READDIRPLUS 请求的数据部分的最大大小。此参数控制服务器返回的请求的最大大小以及客户机生成的请求的最大大小。

数据类型 整数（32 位）

缺省值 1048576 字节

范围 0 到 2^{31} - 1

单位 字节

是否为动态 是，但此参数是在挂载每个文件系统时设置的。要影响特定文件系统，请在更改此参数之后取消挂载然后再挂载该文件系统。

验证 无。但是，如果将服务器上的最大传输大小设置为 0，则可能会导致客户机无法正常运行，或者导致客户机决定不尝试与服务器进行对话。

何时更改 请勿更改此参数，除非首选传输大小大于 1 MB。

承诺级别 不稳定

nfssrv 模块参数

本节介绍 nfssrv 模块的 NFS 参数。

nfssrv:nfs_portmon

说明	控制 NFS 服务器为了强制实施部分客户机上的完整性而尝试执行的某些安全检查。NFS 服务器可以检查发送请求的源端口是否为**保留端口**。保留端口的编号小于 1024。对于基于 BSD 的系统，这些端口是为由超级用户运行的进程保留的。此安全检查可以防止用户编写他们自己的基于 RPC 的应用程序，这样的应用程序会使 NFS 客户机使用的访问检查无法执行。
数据类型	整数（32 位）
缺省值	0（禁用安全检查）
范围	0（禁用安全检查）或 1（启用安全检查）
单位	布尔值
是否为动态	是
验证	无
何时更改	使用此参数防止恶意用户使用 NFS 服务器获取对其通常无法访问的文件的访问权限。但是，**保留端口**概念还没有得到普遍支持。因此，安全检查方面还很薄弱。此外，并非所有的 NFS 客户机实现都将其传输端点绑定到保留范围内的端口号。因此，如果启用安全检查，则可能会导致互操作性问题。
承诺级别	不稳定

nfssrv:rfs_write_async

说明	控制 NFS 版本 2 服务器处理 WRITE 请求时的行为。NFS 版本 2 协议要求在服务器可以响应客户机之前，与 WRITE 请求关联的所有已修改数据和元数据都驻留在稳定存储器上。NFS 版本 2 WRITE 请求最多可包含 8192 字节数据。因此，每个 WRITE 请求都可能会导致向存储子系统执行多个小型写入。这可能会引发性能问题。 一种加速 NFS 版本 2 WRITE 请求的方法是利用客户机行为。客户机可能会以批处理模式发送 WRITE 请求。服务器可以利用此行为，方法是将多个不同的 WRITE 请求群集为单个发往基础文件系统的请求。因此，为数不多的更大请求便可以写入将要写入存储子系统的数据。此方法可以显著增加 WRITE 请求的吞吐量。
数据类型	整数（32 位）

缺省值	1（启用群集）
范围	0（禁用群集）或 1（启用群集）
单位	布尔值
是否为动态	是
验证	无
何时更改	某些很小的 NFS 客户机（特别是 PC 客户机）可能无法批处理 WRITE 请求。因此，所需的客户机行为可能不存在。此外，在 NFS 版本 2 服务器中进行群集操作可能只会增加开销并降低性能，而不会提高性能。
承诺级别	不稳定

nfssrv:nfsauth_ch_cache_max

说明	控制与 NFS 验证服务器进行联系的客户机句柄的高速缓存大小。此服务器验证 NFS 客户机以确定是否允许这些客户机访问它们要尝试使用的文件句柄。
数据类型	整数（32 位）
缺省值	16
范围	0 到 2^{31} - 1
单位	客户机句柄
是否为动态	是
验证	无
何时更改	此高速缓存不是动态的，因此在所有客户机句柄已用完的情况下分配客户机句柄的尝试将失败。此失败会导致 NFS 服务器因无法对这些请求进行验证而丢弃请求。通常，此结果并不重要，因为 NFS 客户机只是超时并且会重新传输请求。但是，对于客户机上的软挂载文件系统，客户机可能会超时，并且不会重试请求，而是将错误返回到应用程序。如果确保服务器上的高速缓存大小足以处理负载，则可以避免这种情况。
承诺级别	不稳定

nfssrv:exi_cache_time

说明	控制在清除项（由于系统内存压力）之前将其保留在 NFS 验证高速缓存中的持续时间。
数据类型	长整数（在 32 位平台上为 32 位，在 64 位平台上为 64 位）

缺省值	3600 秒（1 小时）
范围	在 32 位平台上，为 0 到 2^{31} - 1 在 64 位平台上，为 0 到 2^{63} - 1
单位	秒
是否为动态	是
验证	无
何时更改	通过改变可从高速缓存中清除的项的最短生命期，可以调整 NFS 验证高速缓存的大小。应该控制高速缓存的大小以禁止其增长过大，从而可以使用由于失效过程而不允许释放的系统资源。
承诺级别	不稳定

rpcmod 模块参数

本节介绍 rpcmod 模块的 NFS 参数。

rpcmod:clnt_max_conns

说明	控制 NFS 客户机与每个 NFS 服务器进行通信时使用的 TCP 连接数。将构造内核 RPC，以便可以通过单个连接多路复用 RPC。但是，如有必要，可以使用多个连接。
数据类型	整数（32 位）
缺省值	1
范围	1 到 2^{31} - 1
单位	连接
是否为动态	是
验证	无
何时更改	通常，一个连接足以实现完整的网络带宽。但是，如果 TCP 不能利用网络以单个流形式提供的带宽，则多个连接可能会增加客户机与服务器之间的吞吐量。 增加连接数会产生一些负面影响。增加连接数还会增加跟踪每个连接所需的内核资源使用量。
承诺级别	不稳定

rpcmod:clnt_idle_timeout

说明	控制允许客户机与服务器之间的连接在关闭之前在客户机上保持空闲的持续时间。
数据类型	长整数（在 32 位平台上为 32 位，在 64 位平台上为 64 位）
缺省值	300,000 毫秒（5 分钟）
范围	在 32 位平台上，为 0 到 2^{31} - 1 在 64 位平台上，为 0 到 2^{63} - 1
单位	毫秒
是否为动态	是
验证	无
何时更改	使用此参数更改允许空闲连接在关闭之前在客户机上存在的时间。可能需要以更快的速度关闭连接以免占用系统资源。
承诺级别	不稳定

rpcmod:svc_idle_timeout

说明	控制允许客户机与服务器之间的连接在关闭之前在服务器上保持空闲的持续时间。
数据类型	长整数（在 32 位平台上为 32 位，在 64 位平台上为 64 位）
缺省值	360,000 毫秒（6 分钟）
范围	在 32 位平台上，为 0 到 2^{31} - 1 在 64 位平台上，为 0 到 2^{63} - 1
单位	毫秒
是否为动态	是
验证	无
何时更改	使用此参数更改允许空闲连接在关闭之前在服务器上存在的时间。可能需要以更快的速度关闭连接以免占用系统资源。
承诺级别	不稳定

rpcmod:svc_default_stksize

说明	为内核 RPC 服务线程设置内核栈的大小。

数据类型　整数（32 位）

缺省值　缺省值为 0。此值意味着将栈大小设置为系统缺省值。

范围　0 到 2^{31} - 1

单位　字节

是否为动态　是，对于所有分配的新线程均为动态。栈大小是在创建线程时设置的。因此，对此参数所做的更改不会影响现有线程，但会应用于所有分配的新线程。

验证　无

何时更改　调用深度过深会导致栈溢出以及红色区域故障。相当深的传输调用深度，再加上深的本地文件系统调用，可能会导致 NFS 服务线程的栈溢出。

将此参数设置为平台上的硬件 pagesize 的倍数。

承诺级别　不稳定

rpcmod:svc_default_max_same_xprt

说明　控制在切换传输端点之前每个传输端点处理的最大请求数。内核 RPC 通过一个服务线程池和一个传输端点池进行工作。任一服务线程都可以处理来自任一传输端点的请求。出于性能考虑，在切换到其他传输端点之前，会处理每个传输端点上的多个请求。此方法可避免某些传输端点得到到处理，同是又可以改善性能。

数据类型　整数（32 位）

缺省值　8

范围　0 到 2^{31} - 1

单位　请求

是否为动态　是，但是在切换传输端点之前要处理的最大请求数是在将传输端点配置到内核 RPC 子系统时设置的。对此参数所做的更改只影响新的传输端点，而不会影响现有传输端点。

验证　无

何时更改　调优此参数，以便服务可以利用客户机行为，例如加速 NFS 版本 2 WRITE 请求的群集。增大此参数可能会使服务器更好地利用客户机行为。

承诺级别　不稳定

rpcmod:maxdupreqs

说明　控制用于检测无连接传输中 RPC 级别重新传输的重复请求高速缓存的大小。按照客户机网络地址、RPC 过程编号、程序编号、版本号以及事务 ID 建立此高速缓存的索引。此高速缓存可避免处理可能非等幂的已重新传输的请求。

数据类型　整数（32 位）

缺省值　1024

范围　1 到 2^{31} - 1

单位　请求

是否为动态　该高速缓存的大小是动态设置的，但是用于提供对该高速缓存的快速访问的散列队列的大小则是静态设置的。将此高速缓存设置过大可能会导致在高速缓存中查找项的搜索时间过长。

请勿将此参数的值设置为 0。此值会防止 NFS 服务器处理非等幂请求。

验证　无

何时更改　如果 NFS 客户机遇到伪故障，请检查此参数的值。例如，如果创建目录的尝试失败，但是实际上已经创建了该目录，则可能是服务器未检测到已重新传输的 MKDIR 请求。

高速缓存的大小应该与服务器上的负载相匹配。高速缓存记录非等幂请求，因此，只需跟踪部分请求。高速缓存需要将信息保存足够长的时间以便客户机可以检测到重新传输。通常，无连接传输的客户机超时相对较短，大约在 1 秒到 20 秒之间。

承诺级别　不稳定

rpcmod:cotsmaxdupreqs

说明　控制用于检测面向连接的传输中 RPC 级别重新传输的重复请求高速缓存的大小。按照客户机网络地址、RPC 过程编号、程序编号、版本号以及事务 ID 建立此高速缓存的索引。此高速缓存可避免处理可能非等幂的已重新传输的请求。

数据类型　整数（32 位）

缺省值　1024

范围　1 到 2^{31} - 1

单位　请求

是否为动态　是

验证

该高速缓存的大小是动态设置的，但是用于提供对该高速缓存的快速访问的散列队列的大小是静态设置的。将此高速缓存设置过大可能会导致在高速缓存中查找项的搜索时间过长。

请勿将此参数的值设置为 0。此值会防止 NFS 服务器处理非等幂请求。

何时更改

如果 NFS 客户机遇到伪故障，请检查此参数的值。例如，如果创建目录的尝试失败，但是实际上已经创建了该目录，则可能是服务器未检测到已重新传输的 MKDIR 请求。

高速缓存的大小应该与服务器上的负载相匹配。高速缓存记录非等幂请求，因此，只需跟踪部分请求。它需要将信息保存足够长的时间以便客户机可以检测到重新传输。通常，面向连接的传输的客户机超时很长，大约为 1 分钟。因此，项需要在高速缓存中保留很长一段时间。

承诺级别

不稳定

◆ ◆ ◆ 第 4 章

Internet 协议套件可调参数

本章介绍各种 Internet 协议套件参数，如 TCP、IP、UDP 以及 SCTP。

- 第 114 页中的 “IP 可调参数”
- 第 119 页中的 “TCP 可调参数”
- 第 132 页中的 “UDP 可调参数”
- 第 133 页中的 “IPQoS 可调参数”
- 第 133 页中的 “SCTP 可调参数”
- 第 141 页中的 “每路由度量标准”

有关可调参数的参考信息

可调参数	参考
Solaris 内核可调参数	第 2 章
NFS 可调参数	第 3 章
网络高速缓存和加速器 (Network Cache and Accelerator, NCA) 可调参数	第 5 章

调优 IP 套件参数概述

有关 IP 转发的最新信息，请参见第 20 页中的 “新增和已更改的 TCP/IP 参数” 。

可以使用 `ndd` 命令设置本章中介绍的所有调优参数，但第 129 页中的 “`ipcl_conn_hash_size`” 除外，此参数只能在 `/etc/system` 文件中设置。

例如，按照以下语法使用 `ndd` 命令可设置 TCP/IP 参数：

```
# ndd -set driver parameter
```

有关更多信息，请参见ndd(1M)。

虽然 SMF 框架提供了一种管理系统服务的方法，但是系统启动脚本中仍然包括 ndd 命令。有关创建启动脚本的更多信息，请参见《System Administration Guide: Basic Administration》中的"Run Control Scripts"。

IP 套件参数验证

本节中介绍的所有参数都会被检验是否处于参数范围内。每个参数的说明中都提供参数的范围。

Internet 请求注解文档 (Request for Comment, RFC)

RFC 文档中介绍了 Internet 协议和标准规范。可以从 ftp://ftp.rfc-editor.org/in-notes 获取 RFC 的副本。通过查看此站点上的 rfc-index.txt 文件可浏览 RFC 主题。

IP 可调参数

ip_icmp_err_interval 和 ip_icmp_err_burst

说明　控制 IP 生成 IPv4 或 IPv6 ICMP 错误消息的速率。IP 在某一 ip_icmp_err_interval 间隔内最多只生成 ip_icmp_err_burst 条 IPv4 或 IPv6 ICMP 错误消息。

ip_icmp_err_interval 参数可防止 IP 受到拒绝服务攻击。将此参数设置为 0 可禁用速率限制。但并不禁用错误消息的生成。

缺省值　对于 ip_icmp_err_interval，为 100 毫秒

对于 ip_icmp_err_burst，为 10 条错误消息

范围　对于 ip_icmp_err_interval，为 0 – 99,999 毫秒

对于 ip_icmp_err_burst，为 1 – 99,999 条错误消息

是否为动态　是

何时更改　为了进行诊断而需要更高的错误消息生成速率时。

承诺级别　不稳定

ip_respond_to_echo_broadcast 和 ip6_respond_to_echo_multicast

说明	控制 IPv4 或 IPv6 是否响应广播 ICMPv4 回显请求或多址广播 ICMPv6 回显请求。
缺省值	1（启用）
范围	0（禁用）或 1（启用）
是否为动态	是
何时更改	如果出于安全原因而不想要此行为，则将其禁用。
承诺级别	不稳定

ip_send_redirects 和 ip6_send_redirects

说明	控制 IPv4 或 IPv6 是否发出 ICMPv4 或 ICMPv6 重定向消息。
缺省值	1（启用）
范围	0（禁用）或 1（启用）
是否为动态	是
何时更改	如果出于安全原因而不想要此行为，则将其禁用。
承诺级别	不稳定

ip_forward_src_routed 和 ip6_forward_src_routed

说明	控制 IPv4 或 IPv6 是否转发带有源 IPv4 路由选项或 IPv6 路由数据包头的包。
缺省值	0（禁用）
范围	0（禁用）或 1（启用）
是否为动态	是
何时更改	保持此参数为禁用状态以防止拒绝服务攻击。
承诺级别	不稳定
更改历史记录	有关信息，请参见第 159 页中的 “ip_forward_src_routed 和 ip6_forward_src_routed (Solaris 10)”。

ip_addrs_per_if

说明　　定义与某个实际接口关联的最大逻辑接口数。

缺省值　　256

范围　　1 到 8192

是否为动态　　是

何时更改　　请勿更改此值。如果需要更多的逻辑接口，则可以考虑增大此值。但是，应该意识到这种更改可能会对 IP 性能产生负面影响。

承诺级别　　不稳定

ip_strict_dst_multihoming 和 ip6_strict_dst_multihoming

说明　　确定对于未在非转发接口上显式配置的 IP 地址是否可以接受到达此接口的包。如果已启用 `ip_forwarding`，或者已针对相应接口启用 *xxx*:`ip_forwarding`，则可以忽略此参数，因为实际上已转发了包。

请参阅 RFC 1122 中的第 3.3.4.2 节。

缺省值　　0（宽松的多宿主）

范围　　0 ＝关闭（宽松的多宿主）

1 ＝打开（严格的多宿主）

是否为动态　　是

何时更改　　如果计算机接口通过严格的网络域（如防火墙或 VPN 节点），则将此参数设置为 1。

承诺级别　　不稳定

ip_multidata_outbound

说明　　使网络栈在传输过程中能够一次向网络设备驱动程序发送多个包。

启用此参数可以通过提高主机 CPU 使用率和/或网络吞吐量来减少每个包的处理开销。

现在，此参数控制使用多数据传输 (multidata transmit, MDT) 来传输 IP 分段。例如，当发出大于链路 MTU 的 UDP 有效负荷时。启用此可调参数

之后，会将特定的高级协议（如 UDP）的 IP 分段成批传送到网络设备驱动程序。如果禁用此功能，则会导致网络栈中的 TCP 和 IP 分段逻辑恢复为每次向驱动程序发送一个包。

MDT 功能仅对支持此功能的设备驱动程序有效。

另请参见第 128 页中的 “tcp_mdt_max_pbufs”。

缺省值 1（启用）

范围 0（禁用）或 1（启用）

是否为动态 是

何时更改 如果出于调试目的或任何其他原因不希望启用此参数，请将其禁用。

承诺级别 不稳定

更改历史记录 有关信息，请参见第 159 页中的 “ip_multidata_outbound (Solaris 10)”。

ip_squeue_worker_wait

说明 控制唤醒工作线程来处理在 squeue 中排队的 TCP/IP 包的最大延迟。*squeue* 是指 TCP/IP 内核代码用来处理 TCP/IP 包的连续队列。

缺省值 10 毫秒

范围 0 – 50 毫秒

是否为动态 是

何时更改 如果对网络的响应及时性要求较高，并且网络通信流量较小，则考虑调优此参数。例如，计算机主要用于处理交互式网络通信时。

通常，在网络文件服务器、Web 服务器或者任何具有大量网络通信流量的服务器上，使用缺省值最合适。

承诺级别 不稳定

ip_squeue_enter

说明 控制 squeue 操作在处理来自应用程序的 TCP/IP 包时的行为。

值为 1 会使运行的线程只处理当前包。值为 2 会使运行的线程处理所有在 squeue 中排队的包。

缺省值 2

范围 0 到 2

是否为动态	是
何时更改	在某些情况下，考虑将此参数更改为 1 以提高某些应用程序的网络性能。例如，当 CPU 数超过活动的 NIC 数时，请将此参数更改为 1。 否则，不要更改此参数。
承诺级别	不稳定
更改历史记录	有关信息，请参见第 161 页中的 “ip_squeue_write（Solaris 10 发行版）”。

ip_squeue_fanout

说明	确定将 TCP/IP 连接与 squeue 关联的模式。 值为 0 时会将新的 TCP/IP 连接与创建此连接的 CPU 进行关联。值为 1 时会将此连接与属于不同 CPU 的多个 squeue 进行关联。用于扇出 (fanout) 连接的 squeue 数基于第 118 页中的 “ip_soft_rings_cnt”。
缺省值	0
范围	0 或 1
是否为动态	是
何时更改	在某些情况下，考虑将此参数设置为 1 以将负载分散到所有 CPU。例如，当 CPU 数超过 NIC 数，并且一个 CPU 不能处理一个 NIC 的网络负载时，请将此参数设置为 1。
承诺级别	不稳定
更改历史记录	有关信息，请参见第 160 页中的 “ip_squeue_fanout (Solaris 10)”。

ip_soft_rings_cnt

说明	确定用于扇出 (fanout) 传入 TCP/IP 连接的 squeue 数。 **注 –** 将传入通信流量放在其中一个环上。如果此环过载，则会丢弃包。每丢弃一个包，kstat dls 计数器 `dls_soft_ring_pkt_drop` 会增加 1。
缺省值	2
范围	0 - n 个 CPU，其中 n 是系统中的最大 CPU 数
是否为动态	否。更改此参数时，应该再次检测接口。
何时更改	在具有 10 Gbps NIC 以及多个 CPU 的系统上，请考虑将此参数设置为大于 2 的值。

承诺级别　过时

带有附加注意事项的 IP 可调参数

建议不要更改以下参数。

ip_ire_pathmtu_interval

说明　指定 IP 刷新路径最大传输单元 (path maximum transfer unit, PMTU) 搜索信息并尝试重新搜索 PMTU 的间隔（以毫秒为单位）。

请参阅有关 PMTU 搜索的 RFC 1191。

缺省值　10 分钟

范围　5 秒到 277 小时

是否为动态　是

何时更改　请勿更改此值。

承诺级别　不稳定

ip_icmp_return_data_bytes 和 ip6_icmp_return_data_bytes

说明　当 IPv4 或 IPv6 发送 ICMPv4 或 ICMPv6 错误消息时，它包括导致此错误消息的 IP 数据包头。此参数控制 ICMPv4 或 ICMPv6 错误消息中包括的超过 IPv4 或 IPv6 数据包头的额外包字节数。

缺省值　64 字节

范围　8 到 65,536 字节

是否为动态　是

何时更改　请勿更改此值。在 ICMP 错误消息中包括更多信息可能有助于诊断网络问题。如果需要此功能，则增大此值。

承诺级别　不稳定

TCP 可调参数

tcp_deferred_ack_interval

说明　指定非直接连接的主机的 TCP 延迟确认 (acknowledgment, ACK) 计时器的超时值。

请参阅 RFC 1122 中的第 4.2.3.2 节。

缺省值　100 毫秒

范围　1 毫秒到 1 分钟

是否为动态　是

何时更改　请勿将此值增大到多于 500 毫秒。

在以下情况下应增大此值：

- 网络链接速度较慢（小于 57.6 Kbps），并且最大段大小 (maximum segment size, MSS) 大于 512 字节
- 接收多个 TCP 段的间隔很短

承诺级别　不稳定

tcp_local_dack_interval

说明　指定直接连接的主机的 TCP 延迟确认 (acknowledgment, ACK) 计时器的超时值。

请参阅 RFC 1122 中的第 4.2.3.2 节。

缺省值　50 毫秒

范围　1 毫秒到 1 分钟

是否为动态　是

何时更改　请勿将此值增大到多于 500 毫秒。

在以下情况下应增大此值：

- 网络链接速度较慢（小于 57.6 Kbps），并且最大段大小 (maximum segment size, MSS) 大于 512 字节
- 接收多个 TCP 段的间隔很短

承诺级别　不稳定

tcp_deferred_acks_max

说明　指定在生成确认 (acknowledgment, ACK) 之前从非直接连接的远程目标接收的最大TCP 段数。TCP 段以单个连接的最大段大小 (maximum segment size, MSS) 为单位。如果设置为 0 或 1，则不延迟 ACK（假设所有段的长度均为 1 MSS）。将动态计算每个连接的实际数量。此值为缺省的最大值。

缺省值　2

范围	0 到 16
是否为动态	是
何时更改	请勿更改此值。在某些情况下，如由于 ACK 延迟致使网络通信流量突然变高时，请减小此值。但不要将此值减小到 2 以下。
承诺级别	不稳定

tcp_local_dacks_max

说明	指定在生成确认 (acknowledgment, ACK) 之前从直接连接的目标接收的最大 TCP 段数。TCP 段以单个连接的最大段大小 (maximum segment size, MSS) 为单位。如果设置为 0 或 1，则表示 ACK 不会发生延迟（假设所有段的长度均为 1 MSS）。将动态计算每个连接的实际数量。此值为缺省的最大值。
缺省值	8
范围	0 到 16
是否为动态	是
何时更改	请勿更改此值。在某些情况下，如由于 ACK 延迟致使网络通信流量突然变高时，请减小此值。但不要将此值减小到 2 以下。
承诺级别	不稳定

tcp_wscale_always

说明	当启用此参数（为缺省设置）时，TCP 始终发送带有窗口缩放比例选项的 SYN 段，即使窗口缩放比例选项值为 0 时也是如此。请注意，如果 TCP 收到带有窗口缩放比例选项的 SYN 段，则即使禁用了此参数，TCP 也会使用带有窗口缩放比例选项的 SYN 段进行响应。此外，应根据接收窗口大小设置此选项值。 请参阅 RFC 1323 以了解窗口缩放比例选项。
缺省值	1（启用）
范围	0（禁用）或 1（启用）
是否为动态	是
何时更改	如果与不支持窗口缩放比例选项的旧版 TCP 协议栈之间存在互操作性问题，则禁用此参数。
承诺级别	不稳定
更改历史记录	有关信息，请参见第 162 页中的 “tcp_wscale_always（Solaris 9 发行版）”。

tcp_tstamp_always

说明	如果设置为 1，则 TCP 将始终发送带有时间标记选项的 SYN 段。请注意，如果 TCP 收到带有时间标记选项的 SYN 段，则即使此参数设置为 0，TCP 也会使用带有时间标记的 SYN 段进行响应。
缺省值	0（禁用）
范围	0（禁用）或 1（启用）
是否为动态	是
何时更改	如果难以准确测量往返时间 (round-trip time, RTT) 并且可能出现 TCP 序列号回绕问题，则启用此参数。 请参阅 RFC 1323 以了解有关要启用此选项的更多理由。
承诺级别	不稳定

tcp_xmit_hiwat

说明	定义缺省的发送窗口大小（以字节为单位）。有关针对每个路由设置不同值的介绍，请参阅第 141 页中的“每路由度量标准”。另请参见第 123 页中的“tcp_max_buf”。
缺省值	49,152
范围	4096 到 1,073,741,824
是否为动态	是
何时更改	应用程序可以使用 setsockopt(3XNET) SO_SNDBUF 来更改单个连接的发送缓冲区。
承诺级别	不稳定

tcp_recv_hiwat

说明	定义缺省的接收窗口大小（以字节为单位）。有关针对每个路由设置不同值的介绍，请参阅第 141 页中的“每路由度量标准”。另请参见第 123 页中的“tcp_max_buf”和第 131 页中的“tcp_recv_hiwat_minmss”。
缺省值	49,152
范围	2048 到 1,073,741,824
是否为动态	是
何时更改	应用程序可以使用 setsockopt(3XNET) SO_RCVBUF 来更改单个连接的接收缓冲区。

承诺级别 不稳定

tcp_max_buf

说明 定义最大缓冲区大小（以字节为单位）。此参数控制由使用 setsockopt(3XNET) 的应用程序设置的发送缓冲区和接收缓冲区的大小。

缺省值 1,048,576

范围 8192 到 1,073,741,824

是否为动态 是

何时更改 如果要在高速网络环境中建立 TCP 连接，则增大此值以便适应较高的网络链接速度。

承诺级别 不稳定

tcp_cwnd_max

说明 定义 TCP 拥塞窗口 (congestion window, cwnd) 的最大值（以字节为单位）。

有关 TCP 拥塞窗口的更多信息，请参阅 RFC 1122 和 RFC 2581。

缺省值 1,048,576

范围 128 到 1,073,741,824

是否为动态 是

何时更改 即使应用程序使用 setsockopt(3XNET) 将窗口大小更改为大于 tcp_cwnd_max 的值，所用的实际窗口仍然不会超过 tcp_cwnd_max。因此，tcp_max_buf 应该大于 tcp_cwnd_max。

承诺级别 不稳定

tcp_slow_start_initial

说明 定义 TCP 连接的最大初始拥塞窗口 (congestion window, cwnd) 大小，以最大段大小 (maximum segment size, MSS) 为单位。

有关如何计算初始拥塞窗口大小，请参阅 RFC 2414。

缺省值 4

范围 1 到 4

是否为动态 是

何时更改　请勿更改此值。

如果初始 cwnd 大小在特定情况下导致网络拥塞，则减小此值。

承诺级别　不稳定

tcp_slow_start_after_idle

说明　TCP 连接在空闲（未接收段）一个重新传输超时 (retransmission timeout, RTO) 时间段之后的拥塞窗口大小，以最大段大小 (maximum segment size, MSS) 为单位。

有关如何计算初始拥塞窗口大小，请参阅 RFC 2414。

缺省值　4

范围　1 到 16,384

是否为动态　是

何时更改　有关更多信息，请参见第 123 页中的 “tcp_slow_start_initial”。

承诺级别　不稳定

tcp_sack_permitted

说明　如果设置为 2，则 TCP 将始终发送带有允许选择性确认 (selective acknowledgment, SACK) 的选项的 SYN 段。如果 TCP 收到带有允许 SACK 的选项的 SYN 段，并且此参数设置为 1，则 TCP 会使用允许 SACK 的选项进行响应。如果此参数设置为 0，则无论传入段是否包含允许 SACK 的选项，TCP 都不会发送允许 SACK 的选项。

有关 SACK 选项的信息，请参阅 RFC 2018。

缺省值　2（主动启用）

范围　0（禁用）、1（被动启用）或 2（主动启用）

是否为动态　是

何时更改　SACK 处理可以提高 TCP 重新传输性能，因此应该将其主动启用。有时，主动启用 SACK 选项会使另一端出现混乱。如果出现混乱，则将此值设置为 1，以便只在传入连接允许 SACK 处理时启用 SACK 处理。

承诺级别　不稳定

tcp_rev_src_routes

说明 如果设置为 0，则出于安全原因 TCP 不会对传入连接的 IP 源路由选项进行反向。如果设置为 1，则 TCP 会正常进行源路由反向。

缺省值 0（禁用）

范围 0（禁用）或 1（启用）

是否为动态 是

何时更改 如果进行诊断时需要 IP 源路由，则将其启用。

承诺级别 不稳定

tcp_time_wait_interval

说明 指定 TCP 连接保持 TIME-WAIT 状态的时间（以毫秒为单位）。

有关更多信息，请参阅 RFC 1122 中的第 4.2.2.13 节。

缺省值 60,000（60 秒）

范围 1 秒到 10 分钟

是否为动态 是

何时更改 请不要将此值设置为低于 60 秒的值。

有关更改此参数的信息，请参阅 RFC 1122 中的第 4.2.2.13 节。

承诺级别 不稳定

tcp_ecn_permitted

说明 控制显式拥塞通知 (Explicit Congestion Notification, ECN) 支持。

如果此参数设置为 0，则 TCP 不会与支持 ECN 机制的对等方进行协商。

如果启动连接时此参数设置为 1，则 TCP 不会告知对等方其支持 ECN 机制。

但是，如果对等方在 SYN 段中指示其支持 ECN 机制，则当接受新的传入连接请求时，TCP 会告知此对等方其支持 ECN 机制。

如果此参数设置为 2，则除了在接受连接时与对等方协商 ECN 机制之外，TCP 还在建立主动传出连接时在传出 SYN 段中指示其支持 ECN 机制。

有关 ECN 的信息，请参阅 RFC 3168。

缺省值　1（被动启用）

范围　0（禁用）、1（被动启用）或 2（主动启用）

是否为动态　是

何时更改　ECN 可以帮助 TCP 更好地处理拥塞控制。但是，此机制会使现有的 TCP 实现、防火墙、NAT 以及其他网络设备出现混乱。这些设备不符合 IETF 标准。

由于存在这些设备，因此将此参数的缺省值设置为 1。在极少数情况下，被动启用仍会导致问题。仅当完全必要时将此参数设置为 0。

承诺级别　不稳定

tcp_conn_req_max_q

说明　指定缺省的 TCP 侦听器等待 accept(3SOCKET) 接受的最大暂挂 TCP 连接数。另请参见第 126 页中的 "tcp_conn_req_max_q0"。

缺省值　128

范围　1 到 4,294,967,296

是否为动态　是

何时更改　对于可能接收多个连接请求的应用程序（如 Web 服务器），可以增大缺省值以便适应较高的传入速率。

请不要将此参数增大到非常大的值。暂挂 TCP 连接可能会占用过多内存。此外，如果由于暂挂 TCP 连接数过多而导致应用程序不能迅速处理如此多的连接请求，则可能会拒绝新的传入请求。

请注意，增大 tcp_conn_req_max_q 并不意味着应用程序可以具有如此多的暂挂 TCP 连接。应用程序可以使用 listen(3SOCKET) 来更改每个套接字的最大暂挂 TCP 连接数。此参数是应用程序可以使用 listen() 设置的最大值。因此，即使根据 listen() 中使用的值将此参数设置为非常大的值，套接字的实际最大数量也可能比 tcp_conn_req_max_q 小得多。

承诺级别　不稳定

更改历史记录　有关信息，请参见第 163 页中的 "*xxx*:ip_forwarding（Solaris 9 发行版）"。

tcp_conn_req_max_q0

说明　指定缺省的 TCP 侦听器未完成的（尚未完成三次握手）最大暂挂 TCP 连接数。

有关 TCP 三次握手的更多信息，请参阅 RFC 793。另请参见第 126 页中的“tcp_conn_req_max_q”。

缺省值　1024

范围　0 到 4,294,967,296

是否为动态　是

何时更改　对于可能收到过多连接请求的应用程序（如 Web 服务器），可以增大缺省值以便适应较高的传入速率。

下面介绍了 tcp_conn_req_max_q0 与每个套接字的最大暂挂连接数之间的关系。

收到连接请求时，TCP 首先检查侦听器等待被接受的暂挂 TCP 连接（已执行三次握手）数目是否超过最大值 (*N*)。如果连接数过多，则会拒绝请求。如果连接数为允许数目，则 TCP 会检查未完成的暂挂 TCP 连接数是否超过 *N* 与 tcp_conn_req_max_q0 的总和。如果没有超过，则接受请求。否则，会删除最早未完成的暂挂 TCP 请求。

承诺级别　不稳定

更改历史记录　有关信息，请参见第 163 页中的“*xxx*:ip_forwarding（Solaris 9 发行版）”。

tcp_conn_req_min

说明　指定缺省的侦听器等待被接受的最大暂挂 TCP 连接请求数的最小值。这是应用程序可以使用的最大 listen(3SOCKET) 值的下限值。

缺省值　1

范围　1 到 1024

是否为动态　是

何时更改　对于使用 listen(3SOCKET) 设置的最大暂挂 TCP 连接数过低的应用程序，此参数可提供一种解决方案。请增大此值以便适应较高的传入连接请求速率。

承诺级别　不稳定

tcp_rst_sent_rate_enabled

说明　如果此参数设置为 1，则 RST 段的最大发送速率受 ndd 参数 tcp_rst_sent_rate 控制。如果此参数设置为 0，则不对 RST 段的发送速率进行任何控制。

缺省值	1（启用）
范围	0（禁用）或 1（启用）
是否为动态	是
何时更改	此可调参数通过限制 RST 段的发出速率来防止 TCP 受到拒绝服务攻击。仅当要求严格遵守 RFC 793 时，才应该禁用此速率控制。
承诺级别	不稳定

tcp_rst_sent_rate

说明	设置 TCP 每秒可以发出的最大 RST 段数。
缺省值	40
范围	0 到 4,294,967,296
是否为动态	是
何时更改	在 TCP 环境中，可能存在合法的理由使生成的 RST 数多于缺省值允许的数量。在这种情况下，请增大此参数的缺省值。
承诺级别	不稳定

tcp_mdt_max_pbufs

说明	指定由 TCP 生成的一条 M_MULTIDATA 消息可以携带的有效负荷缓冲区数。另请参见第 116 页中的 "ip_multidata_outbound"。
缺省值	16
范围	1 到 16
是否为动态	是
何时更改	减小此参数时，可以通过限制由 TCP 生成的每条 M_MULTIDATA 消息携带的有效负荷缓冲区数而有助于调试设备驱动程序开发。
承诺级别	不稳定

在 /etc/system 文件中设置的 TCP/IP 参数

ipcl_conn_hash_size 参数只能在 /etc/system 文件中设置。修改此文件之后，请重新引导系统。

以下项设置 ipcl_conn_hash_size 参数：

```
set ip:ipcl_conn_hash_sizes=value
```

ipcl_conn_hash_size

说明 控制 IP 使用的连接散列表的大小。缺省值 0 意味着系统在引导时根据可用内存自动将此参数调整为相应值。

数据类型 无符号整数

缺省值 0

范围 0 到 82,500

是否为动态 否。只能在引导系统时更改此参数。

何时更改 如果系统始终具有数万个 TCP 连接，则可以相应地增大此值。增加散列表大小意味着占用更多内存，从而减少了用户应用程序的可用内存。

承诺级别 不稳定

带有附加注意事项的 TCP 参数

建议不要更改以下参数。

tcp_ip_abort_interval

说明 指定缺省的 TCP 连接重新传输总超时值。对于给定的 TCP 连接，如果 TCP 包已经重新传输了 `tcp_ip_abort_interval` 时间，但在这段时间内未收到来自另一个端点的任何确认，则 TCP 将关闭此连接。

有关 TCP 重新传输超时 (retransmission timeout, RTO) 计算，请参阅 RFC 1122 中的第 4.2.3 节。另请参见第 130 页中的“tcp_rexmit_interval_max”。

缺省值 8 分钟

范围 500 毫秒到 1193 小时

是否为动态 是

何时更改 请勿更改此值。有关例外，请参见第 130 页中的“tcp_rexmit_interval_max”。

承诺级别 不稳定

tcp_rexmit_interval_initial

说明 指定缺省的 TCP 连接初始重新传输超时 (retransmission timeout, RTO) 值。有关针对每个路由设置不同值的介绍，请参阅第 141 页中的“每路由度量标准”。

缺省值	3 秒
范围	1 毫秒到 20 秒
是否为动态	是
何时更改	请勿更改此值。减小此值可能会导致不必要的重新传输。
承诺级别	不稳定

tcp_rexmit_interval_max

说明　定义缺省的最大重新传输超时 (retransmission timeout, RTO) 值。对于所有 TCP 连接，计算后得出的 RTO 不能超过此值。另请参见第 129 页中的“tcp_ip_abort_interval”。

缺省值　60 秒

范围　1 毫秒到 2 小时

是否为动态　是

何时更改　在一般的网络环境中，请勿更改此值。

在某些特殊情况下，如果连接的往返时间 (round-trip time, RTT) 大约为 10 秒，则可以增大此值。如果更改此值，则还应该更改 tcp_ip_abort_interval 参数。将tcp_ip_abort_interval 的值更改为至少是 tcp_rexmit_interval_max 值的四倍。

承诺级别　不稳定

tcp_rexmit_interval_min

说明　指定缺省的最小重新传输超时 (retransmission timeout, RTO) 值。所有 TCP 连接的已计算的 RTO 不能低于此值。另请参见第 130 页中的“tcp_rexmit_interval_max”。

缺省值　400 毫秒

范围　1 毫秒到 20 秒

是否为动态　是

何时更改　在一般的网络环境中，请勿更改此值。

TCP 的 RTO 计算应该可以应对大多数 RTT 浮动。在某些非常特殊的情况下，如果连接的往返时间 (round-trip time, RTT) 大约为 10 秒，则增大此值。如果更改此值，则应该更改 tcp_rexmit_interval_max 参数。应将 tcp_rexmit_interval_max 的值更改为至少是 tcp_rexmit_interval_min 值的八倍。

承诺级别　不稳定

tcp_rexmit_interval_extra

说明 指定向已计算的重新传输超时 (retransmission time out, RTO) 值中添加的常数。

缺省值 0 毫秒

范围 0 到 2 小时

是否为动态 是

何时更改 请勿更改此值。

当 RTO 计算无法获得连接的最佳值时，可以更改此值以免不必要的重新传输。

承诺级别 不稳定

tcp_tstamp_if_wscale

说明 如果此参数设置为 1，并且针对连接启用窗口缩放比例选项，则 TCP 还将针对此连接启用 `timestamp` 选项。

缺省值 1（启用）

范围 0（禁用）或 1（启用）

是否为动态 是

何时更改 请勿更改此值。通常，当在高速网络中使用 TCP 时，必须防止序列号回绕。因此，需要 `timestamp` 选项。

承诺级别 不稳定

tcp_recv_hiwat_minmss

说明 控制缺省的最小接收窗口大小。最小值是连接的最大段大小 (maximum segment size, MSS) 的 `tcp_recv_hiwat_minmss` 倍。

缺省值 4

范围 1 到 65,536

是否为动态 是

何时更改 请勿更改此值。如果需要更改此值，则不要将其更改为低于 4 的值。

承诺级别 不稳定

UDP 可调参数

udp_xmit_hiwat

说明	定义缺省的最大 UDP 套接字数据报大小。有关更多信息，请参见第 132 页中的 "udp_max_buf"。
缺省值	57,344 字节
范围	1,024 到 1,073,741,824 字节
是否为动态	是
何时更改	请注意，应用程序可以使用 setsockopt(3XNET) SO_SNDBUF 来更改单个套接字的大小。通常情况下，不需要更改缺省值。
承诺级别	不稳定
更改历史记录	有关信息，请参见第 164 页中的 "udp_xmit_hiwat（Solaris 9 发行版）"。

udp_recv_hiwat

说明	定义缺省的最大 UDP 套接字接收缓冲区大小。有关更多信息，请参见第 132 页中的 "udp_max_buf"。
缺省值	57,344 字节
范围	128 到 1,073,741,824 字节
是否为动态	是
何时更改	请注意，应用程序可以使用 setsockopt(3XNET) SO_RCVBUF 来更改单个套接字的大小。通常，不需要更改缺省值。
承诺级别	不稳定
更改历史记录	有关信息，请参见第 164 页中的 "udp_recv_hiwat（Solaris 9 发行版）"。

带有附加注意事项的 UCP 参数

建议不要更改以下参数。

udp_max_buf

说明	控制 UDP 套接字的发送缓冲区和接收缓冲区的大小。

缺省值　　2,097,152 字节
范围　　65,536 到 1,073,741,824 字节

是否为动态　　是

何时更改　　请勿更改此值。如果将此参数设置为非常大的值，则 UDP 套接字应用程序可能会占用过多的内存。

承诺级别　　不稳定

更改历史记录　　有关信息，请参见第 164 页中的 “udp_max_buf（Solaris 9 发行版）” 。

IPQoS 可调参数

ip_policy_mask

说明　　在以下任一标注位置中启用或禁用 IPQoS 处理：转发外发、转发传入、本地外发以及本地传入。此参数是如下所示的位掩码：

未使用	未使用	未使用	未使用	转发外发	转发传入	本地外发	本地传入
X	X	X	X	0	0	0	0

任一位置中的 1 将掩盖或禁用此特定标注位置中的 IPQoS 处理。例如，值 `0x01` 将针对所有本地传入包禁用 IPQoS 处理。

缺省值　　缺省值为 0，意味着在所有标注位置中启用 IPQoS 处理。

范围　　0 (0x00) 到 15 (0x0F)。值 15 指示在所有标注位置中禁用 IPQoS 处理。

是否为动态　　是

何时更改　　要在任一标注位置中启用或禁用 IPQoS 处理时。

承诺级别　　不稳定

SCTP 可调参数

sctp_max_init_retr

说明　　控制 SCTP 端点在重新发送 INIT 块时应该进行的最大尝试次数。SCTP 端点可以使用 SCTP 启动结构覆盖此值。

缺省值　8
范围　0 到 128

是否为动态　是

何时更改　INIT 的重新传输数取决于第 134 页中的 “sctp_pa_max_retr”。理论上，sctp_max_init_retr 应该小于或等于 sctp_pa_max_retr。

承诺级别　不稳定

sctp_pa_max_retr

说明　控制 SCTP 关联的最大重新传输数（所有路径上的）。当超过此数值时，将中止 SCTP 关联。

缺省值　10
范围　1 到 128

是否为动态　是

何时更改　所有路径上的最大重新传输数取决于路径数以及每条路径上的最大重新传输数。理论上，应该将 sctp_pa_max_retr 设置为所有可用路径上的 sctp_pp_max_retr 之和（参见第 134 页中的 “sctp_pp_max_retr”）。例如，如果有 3 条通往目标的路径，并且这 3 条路径中每条路径上的最大重新传输数均为 5，则应该将 sctp_pa_max_retr 设置为小于或等于 15。（请参见 RFC 2960 的第 8.2 节中的注释。）

承诺级别　不稳定

sctp_pp_max_retr

说明　控制特定路径上的最大重新传输数。如果某条路径的重新传输数超过此数值，请将此路径（目标）视为不可访问。

缺省值　5
范围　1 到 128

是否为动态　是

何时更改　请不要将此值更改为小于 5 的值。

承诺级别　不稳定

sctp_cwnd_max

说明　控制 SCTP 关联的最大拥塞窗口值。

缺省值　1,048,576

范围　128 到 1,073,741,824

是否为动态　是

何时更改　即使应用程序使用 setsockopt(3XNET) 将窗口大小更改为大于 sctp_cwnd_max 的值，所用的实际窗口仍然不会超过 sctp_cwnd_max。因此，第 138 页中的 “sctp_max_buf” 应该大于 sctp_cwnd_max。

承诺级别　不稳定

sctp_ipv4_ttl

说明　控制 SCTP 关联的外发 IPv4 包的 IPv4 数据包头中的生存时间 (time to live, TTL) 值。

缺省值　64
范围　1 到 255

是否为动态　是

何时更改　该值通常不需要更改。如果通往目标的路径可能跨越 64 个以上跃点，则考虑增大此参数。

承诺级别　不稳定

sctp_heartbeat_interval

说明　计算向空闲目标发送 HEARTBEAT 块的时间间隔，该间隔为心跳时间。

SCTP 端点定期发送 HEARTBEAT 块，以监视其对等方的空闲目标传输地址的可访问性。

缺省值　30 秒
范围　0 到 86,400 秒

是否为动态　是

何时更改　请参阅 RFC 2960 的第 8.3 节。

承诺级别　不稳定

sctp_new_secret_interval

说明　确定何时需要生成新的机密。生成的机密用于计算 cookie 的 MAC。

缺省值　2 分钟

范围	0 到 1,440 分钟
是否为动态	是
何时更改	请参阅 RFC 2960 的第 5.1.3 节。
承诺级别	不稳定

sctp_initial_mtu

说明	确定包括 IP 数据包头长度的 SCTP 包的初始最大发送大小。
缺省值	1500 字节
范围	68 到 65,535
是否为动态	是
何时更改	如果基础链接支持大于 1500 字节的帧大小，则增大此参数。
承诺级别	不稳定

sctp_deferred_ack_interval

说明	设置 SCTP 延迟确认 (acknowledgment, ACK) 计时器的超时值（以毫秒为单位）。
缺省值	100 毫秒
范围	1 到 60,000 毫秒
是否为动态	是
何时更改	请参阅 RFC 2960 的第 6.2 节。
承诺级别	不稳定

sctp_ignore_path_mtu

说明	启用或禁用路径 MTU 搜索。
缺省值	0（禁用）
范围	0（禁用）或 1（启用）
是否为动态	是
何时更改	如果要忽略路径上的 MTU 更改，则启用此参数。但是，如果路径 MTU 减少，则启用此参数可能会导致 IP 分段。
承诺级别	不稳定

sctp_initial_ssthresh

说明　设置对等方目标地址的初始慢速启动阈值。

缺省值　102,400
范围　1024 到 4,294,967,295

是否为动态　是

何时更改　请参阅 RFC 2960 的第 7.2.1 节。

承诺级别　不稳定

sctp_xmit_hiwat

说明　设置缺省的发送窗口大小（以字节为单位）。另请参见第 138 页中的“sctp_max_buf”。

缺省值　102,400
范围　8,192 到 1,073,741,824

是否为动态　是

何时更改　应用程序可以使用 setsockopt(3SOCKET) SO_SNDBUF 来更改单个关联的发送缓冲区。

承诺级别　不稳定

sctp_xmit_lowat

说明　控制发送窗口大小的下限。

缺省值　8,192
范围　8,192 到 1,073,741,824

是否为动态　是

何时更改　该值通常不需要更改。此参数设置为了将套接字标记为可写而需要在发送缓冲区中包含的最小大小。如果需要，请考虑根据第 137 页中的“sctp_xmit_hiwat”更改此参数。

承诺级别　不稳定

sctp_recv_hiwat

说明　控制缺省的接收窗口大小（以字节为单位）。另请参见第 138 页中的“sctp_max_buf”。

缺省值　102,400
范围　8,192 到 1,073,741,824

是否为动态　是

何时更改　应用程序可以使用 setsockopt(3SOCKET) SO_RCVBUF 来更改单个关联的接收缓冲区。

承诺级别　不稳定

sctp_max_buf

说明　控制最大缓冲区大小（以字节为单位）。它控制由使用 setsockopt(3SOCKET) 的应用程序设置的发送缓冲区和接收缓冲区的大小。

缺省值　1,048,576
范围　8,192 到 1,073,741,824

是否为动态　是

何时更改　如果要在高速网络环境中进行关联，则增大此参数的值以便适应较高的网络链接速度。

承诺级别　不稳定

sctp_ipv6_hoplimit

说明　设置 SCTP 关联的外发 IPv6 包的 IPv6 数据包头中的跃点限制值。

缺省值　60
范围　0 到 255

是否为动态　是

何时更改　该值通常不需要更改。如果通往目标的路径可能跨越 60 个以上跃点，则考虑增大此参数。

承诺级别　不稳定

sctp_rto_min

说明　设置对等方的所有目标地址的重新传输超时 (retransmission timeout, RTO) 下限（以毫秒为单位）。

缺省值　1,000
范围　500 到 60,000

是否为动态 是

何时更改 请参阅 RFC 2960 的第 6.3.1 节。

承诺级别 不稳定

sctp_rto_max

说明 控制对等方的所有目标地址的重新传输超时 (retransmission timeout, RTO) 上限（以毫秒为单位）。

缺省值 60,000
范围 1,000 到 60,000,000

是否为动态 是

何时更改 请参阅 RFC 2960 的第 6.3.1 节。

承诺级别 不稳定

sctp_rto_initial

说明 控制对等方的所有目标地址的初始重新传输超时 (retransmission timeout, RTO)（以毫秒为单位）。

缺省值 3,000
范围 1,000 到 60,000,000

是否为动态 是

何时更改 请参阅 RFC 2960 的第 6.3.1 节。

承诺级别 不稳定

sctp_cookie_life

说明 设置 cookie 的寿命（以毫秒为单位）。

缺省值 60,000
范围 10 到 60,000,000

是否为动态 是

何时更改 该值通常不需要更改。可以根据第 139 页中的 “sctp_rto_max”更改此参数。

承诺级别 不稳定

sctp_max_in_streams

说明 控制 SCTP 关联允许的最大传入流数。

缺省值 32
范围 1 到 65,535

是否为动态 是

何时更改 请参阅 RFC 2960 的第 5.1.1 节。

承诺级别 不稳定

sctp_initial_out_streams

说明 控制 SCTP 关联允许的最大外发流数。

缺省值 32
范围 1 到 65,535

是否为动态 是

何时更改 请参阅 RFC 2960 的第 5.1.1 节。

承诺级别 不稳定

sctp_shutack_wait_bound

说明 控制发送 SHUTDOWN 块之后等待 SHUTDOWN ACK 的最长时间（以毫秒为单位）。

缺省值 60,000
范围 0 到 300,000

是否为动态 是

何时更改 该值通常不需要更改。可以根据第 139 页中的 “sctp_rto_max”更改此参数。

承诺级别 不稳定

sctp_maxburst

说明 设置一次成组传送中发送的段数限制。

缺省值 4
范围 2 到 8

是否为动态　是

何时更改　不需要更改此参数，但可能为了进行测试而更改。

承诺级别　不稳定

sctp_addip_enabled

说明　启用或禁用 SCTP 动态地址重新配置。

缺省值　0（禁用）
范围　0（禁用）或 1（启用）

是否为动态　是

何时更改　如果需要动态地址重新配置，则可以启用此参数。出于安全考虑，只在进行测试时启用此参数。

承诺级别　不稳定

sctp_prsctp_enabled

说明　启用或禁用 SCTP 的部分可靠性扩展 (RFC 3758)。

缺省值　1（启用）
范围　0（禁用）或 1（启用）

是否为动态　是

何时更改　如果 SCTP 环境不支持部分可靠性，则禁用此参数。

承诺级别　不稳定

每路由度量标准

从 Solaris 8 发行版开始，可以使用每路由度量标准将某些属性与 IPv4 和 IPv6 路由表项进行关联。

例如，有一个系统配备了两个不同的网络接口：一个快速以太网接口和一个千兆位以太网接口。系统的缺省 tcp_recv_hiwat 为 24,576 字节。此缺省值对于快速以太网接口是足够了，但可能无法满足千兆位以太网卡接口的需求。

可以将不同的缺省 TCP 接收窗口大小关联到千兆位以太网接口路由项，而不是增大系统的 tcp_recv_hiwat 缺省值。通过进行此关联，所有通过此路由的 TCP 连接都将具有增大的接收窗口大小。

例如，以下是路由表 (netstat -rn) 中的项（采用 IPv4）：

```
192.123.123.0        192.123.123.4        U        1      4  hme0

192.123.124.0        192.123.124.4        U        1      4  ge0

default              192.123.123.1        UG       1      8
```

在本示例中，执行以下操作：

```
# route change -net 192.123.124.0 -recvpipe x
```

然后，所有通往 ge0 链接上的 192.123.124.0 网络的连接都将使用接收缓冲区大小 *x*，而不使用缺省的接收窗口大小 24567。

如果目标位于 a.b.c.d 网络中，并且不存在特定于此网络的路由项，则可以向此网络中添加前缀路由并更改度量标准。例如：

```
# route add -net a.b.c.d 192.123.123.1 -netmask w.x.y.z

# route change -net a.b.c.d -recvpipe y
```

请注意，此前缀路由的网关是缺省路由器。然后，所有通往此网络的连接都使用接收缓冲区大小 *y*。如果有多个接口，则使用 -ifp 参数指定要使用的接口。这样，可以控制将哪个接口用于特定的目标。要检验此度量标准，请使用 route(1M) get 命令。

◆ ◆ ◆ 第 5 章

网络高速缓存和加速器可调参数

本章介绍一些网络高速缓存和加速器 (Network Cache and Accelerator, NCA) 可调参数。

有关可调参数的参考信息

可调参数	参考
Solaris 内核可调参数	第 2 章
NFS 可调参数	第 3 章
Internet 协议套件可调参数	第 4 章

调优 NCA 参数

适合在作为专用 Web 服务器的系统上设置这些参数。这些参数将为高速缓存页面分配更多内存。可以在 /etc/system 文件中设置本章中介绍的所有调优参数。

有关向 /etc/system 文件中添加可调参数的信息，请参见第 24 页中的 “调优 Solaris 内核”。

nca:nca_conn_hash_size

说明	针对所有 TCP 连接控制 NCA 模块中的散列表大小，将其调整为最近的素数。
缺省值	383 个散列表项
范围	0 到 201,326,557
是否为动态	否
何时更改	NCA 的 TCP 散列表太小，无法跟踪传入 TCP 连接时。这种情况会导致将许多 TCP 连接组合到同一散列表项中。当 NCA 接收许多 TCP 连接时会指示此情况，并且系统性能会降低。
承诺级别	不稳定

nca:nca_conn_req_max_q

说明	定义 NCA 可侦听的最大暂挂 TCP 连接数。
缺省值	256 个连接
范围	0 到 4,294,967,295
是否为动态	否
何时更改	NCA 在建立连接后立即将其关闭，因为它已经具有太多已建立的 TCP 连接。如果 NCA 接收许多 TCP 连接并且可以处理较大负载，但是拒绝更多连接，则增大此参数。这样，NCA 便可处理更多的同步 TCP 连接。
承诺级别	不稳定

nca:nca_conn_req_max_q0

说明	针对 NCA 侦听定义最大未完成（尚未完成三次握手）暂挂 TCP 连接数。
缺省值	1024 个连接
范围	0 到 4,294,967,295
是否为动态	否
何时更改	NCA 拒绝接受任何更多的 TCP 连接，因为它已经具有太多的暂挂 TCP 连接。如果 NCA 接收许多 TCP 连接并且可以处理较大负载，但是拒绝更多连接，则增大此参数。这样，NCA 便可处理更多的同步 TCP 连接。
承诺级别	不稳定

nca:nca_ppmax

说明	指定NCA用于高速缓存页面的最大物理内存量（以页为单位）。此值不应超过总内存的75%。
缺省值	物理内存的25%
范围	物理内存的1%到最大物理内存量
是否为动态	否
何时更改	在内存大于512 MB的系统上使用NCA时。如果系统具有大量未使用的物理内存，则增大此参数。然后，NCA将有效地使用此内存来高速缓存新对象。这样，系统性能将得到改善。 应将此参数与nca_vpmax一起增大，除非系统的物理内存大于虚拟内存（内存超过4 GB的32位内核）。使用pagesize(1)确定系统的页面大小。
承诺级别	不稳定

nca:nca_vpmax

说明	指定NCA用于高速缓存页面的最大虚拟内存量（以页为单位）。此值不应超过总内存的75%。
缺省值	虚拟内存的25%
范围	虚拟内存的1%到最大虚拟内存量
是否为动态	否
何时更改	在内存大于512 MB的系统上使用NCA时。如果系统具有大量未使用的虚拟内存，则增大此参数。然后，NCA将有效地使用此内存来高速缓存新对象。这样，系统性能将得到改善。 应将此参数与nca_ppmax一起增大。将此参数设置为几乎与nca_vpmax值相同，除非系统的物理内存大于虚拟内存。
承诺级别	不稳定

NCA的常规系统调优

除了设置NCA参数之外，还可以执行某些常规系统调优以改善NCA性能。如果使用千兆位以太网（ge驱动程序），则应该在中断模式下设置接口以获得更好结果。

例如，对于在64位内核下引导的具有4 GB内存的系统，应该在/etc/system文件中设置以下参数。使用pagesize确定系统的页面大小。

```
set sq_max_size=0

set ge:ge_intr_mode=1

set nca:nca_conn_hash_size=82500

set nca:nca_conn_req_max_q=100000

set nca:nca_conn_req_max_q0=100000

set nca:nca_ppmax=393216

set nca:nca_vpmax=393216
```

sq_max_size

说明　在目标 STREAMS 队列生成 QFULL 消息之前，设置 syncq 的深度（消息数）。

缺省值　10000 条消息

范围　0（没有限制）到 MAXINT

是否为动态　否

何时更改　当 NCA 在高内存的系统上运行时，请增大此参数以允许驱动程序对更多的数据包进行排队。如果服务器为重负载，则增大此参数，以便模块和驱动程序可以处理更多数据，而无需丢弃包或囤积包。

Solaris 8 以及更高发行版－请不要在生产系统上将此参数设置为 0。如果需要更改此参数，则逐步增大此值，并且监视系统。

Solaris 7 以及早期发行版－请不要在生产系统上将此参数设置为 0。如果需要更改此参数，则逐步将此值增大到最大 100，并且监视系统。

承诺级别　不稳定

更改历史记录　有关信息，请参见第 165 页中的 “sq_max_size（Solaris 9 12/02 发行版）”。

ge:ge_intr_mode

说明　启用 ge 驱动程序以直接将包发送到上层通信层，而不是对包进行排队。

缺省值　0（将包排队到上层）

范围　0（启用）或 1（禁用）

是否为动态	否
何时更改	启用NCA时，将此参数设置为1，以便在中断模式下将包传送到NCA以进行更快处理。
承诺级别	不稳定

第 6 章

系统功能参数

本章介绍各种系统功能的多数参数缺省值。

系统缺省参数

各种系统功能的运行由每项功能在启动时读取的一组值进行控制。存储在每项功能的文件中的值位于 /etc/default 目录中。不是每项系统功能都具有位于此目录中的文件。

autofs

利用此功能，可以配置 autofs 参数，如自动超时、显示或记录状态消息、浏览 autofs 挂载点以及跟踪。有关详细信息，请参见 autofs(4)。

cron

利用此功能，可以禁用或启用 cron 日志。

devfsadm

当前未使用此文件。

dhcpagent

dhcpagent 守护进程提供 DHCP 的客户机使用情况。当 ifconfig 识别到已配置为从 DHCP 接收其网络配置的接口时，会启动客户机守护进程来管理此接口。

有关更多信息，请参见 dhcpagent(1M) 的 FILES 部分中的 /etc/default/dhcpagent 信息。

fs

文件系统管理命令具有一个通用的文件系统特定部分。如果未使用 -F 选项显式指定文件系统类型，则应用缺省类型。在此文件中指定值。有关更多信息，请参见 default_fs(4) 的 DESCRIPTION 部分。

ftp

利用此功能，可以将 ls 命令行为设置为 RFC 959 NLST 命令。缺省的 ls 行为与上一个 Solaris 发行版中的行为相同。

有关详细信息，请参见 ftp(4)。

inetinit

利用此功能，可以配置 TCP 序列号，并且可以启用或禁用对 6to4 中继路由器的支持。

init

有关详细信息，请参见 init(1M) 的 FILES 部分中的 /etc/default/init 信息。

文件中的所有值都放置在 init 作为对单用户引导请求的响应而调用的 shell 环境中。init 进程还将这些值传递到从 /etc/inittab 文件启动或重新启动的任何命令。

keyserv

有关详细信息，请参见 keyserv(1M) 的 FILES 部分中的 /etc/default/keyserv 信息。

kbd

有关详细信息，请参见 kbd(1) 的 EXTENDED DESCRIPTION 部分。

login

有关详细信息，请参见 login(1) 的 FILES 部分中的 /etc/default/login 信息。

mpathd

利用此功能，可以设置 in.mpathd 配置参数。

有关详细信息，请参见 in.mpathd(1M)。

nfs

利用此功能，可以设置 NFS 守护进程配置参数。

有关详细信息，请参见 nfs(4)。

nfslogd

有关详细信息，请参见 nfslogd(1M) 的 DESCRIPTION 部分。

nss

利用此功能，可以配置 initgroups(3C) 查找参数。

有关详细信息，请参见 nss(4)。

passwd

有关详细信息，请参见 passwd(1) 的 FILES 部分中的 /etc/default/passwd 信息。

power

有关详细信息，请参见 pmconfig(1M) 的 FILES 部分中的 /etc/default/power 信息。

rpc.nisd

有关详细信息，请参见 rpc.nisd(1M) 的 FILES 部分中的 /etc/default/rpc.nisd 信息。

su

有关详细信息，请参见 su(1M) 的 FILES 部分中的 /etc/default/su 信息。

syslog

有关详细信息，请参见 syslogd(1M) 的 FILES 部分中的 /etc/default/syslogd 信息。

sys-suspend

有关详细信息，请参见 sys-suspend(1M) 的 FILES 部分中的 /etc/default/sys-suspend 信息。

tar

有关 -f 功能修饰符的说明，请参见 tar(1)。

如果 TAPE 环境变量不存在，其中一个参数的值是一个数字，并且未指定 -f，则在 /etc/default/tar 文件中查找与 archive*N* 字符串匹配的数字。将 archive*N* 字符串的值用作具有文件分块和大小规格的输出设备。

例如：

```
% tar -c 2 /tmp/*
```

此命令将输出写入在 /etc/default/tar 文件中指定为 archive2 的设备。

utmpd

utmpd 守护进程将监视 /var/adm/utmpx（以及早期 Solaris 版本中的 /var/adm/utmp），以确保进程终止时清除非根进程通过 pututxline(3C) 插入的 utmp 项。

/etc/default/utmpd 中支持两项：

- SCAN_PERIOD－utmpd 执行下一次 /proc 检查以查看监视的进程是否仍处于活动状态之前休眠的秒数。缺省值为 300。
- MAX_FDS－utmpd 尝试监视的最大进程数。缺省值为 4096 并且永远无需更改。

webconsole

利用此功能，可以配置 Java Web Console 参数。有关更多信息，请参见《System Administration Guide: Basic Administration》中的"Configuring the Java Web Console"。

yppasswdd

利用此功能，可以配置在使用 passwd -r nis -e 命令时用户是否可以成功将登录 shell 设置为受限 shell。

有关详细信息，请参见 rpc.yppasswdd(1M)。

附 录 A

可调参数更改历史记录

本章介绍特定可调参数的更改历史记录。本节中列出的参数与早期发行版中的参数有所不同。另外，还列出了功能已删除的参数。

内核参数

进程大小可调参数

max_nprocs（**Solaris 9 发行版**）

通过删除文本 "sun4m" 更新了 Solaris 10 说明部分。

常规 I/O 可调参数

maxphys（**Solaris 9 发行版**）

Solaris 10 maxphys 缺省值部分中删除了文本 "126,976 (sun4m)"。

rlim_fd_max（**Solaris 8 发行版**）

在 Solaris 8 版本中，缺省值为 1024。在更高的 Solaris 发行版中，缺省值为 65,536。

常规内核参数和内存参数

`lwp_default_stksize`（**Solaris 9 发行版**）

通过添加 AMD64 的缺省值和最大值更新了 Solaris 10 说明部分。

Solaris 10 的 SPARC 平台缺省值已更改为 24,576。

`noexec_user_stack`（**Solaris 9 发行版**）

通过删除文本 "and sun4m" 并添加文本 "64–bit SPARC and AMD64" 更新了 Solaris 10 说明部分。

`noexec_user_stack`（**Solaris 2.6、7 和 8 发行版**）

通过删除文本 "and sun4d" 更新了 Solaris 9 说明部分。

`segkpsize`（**Solaris 9 12/02 发行版**）

在早期的 Solaris 9 发行版中，将单位错误地标识为 MB 而不是 8 KB 页。此外，在 Solaris 10 发行版的范围和缺省值说明中删除了以下文本，因为此参数只能用于运行 64 位内核的系统：

32-bit kernels, 512 Mbytes

`logevent_max_q_sz`（**Solaris 9 发行版**）

在 Solaris 10 发行版中，此参数的缺省值从 2000 个事件更改为 5000 个事件。有关更多信息，请参见第 32 页中的 “`logevent_max_q_sz`”。

与分页相关的可调参数

`tmpfs:tmpfs_minfree`（**Solaris 8 发行版**）

在 Solaris 8 版本中，将单位错误地描述为“字节”而不是“页”。

`pages_pp_maximum`（**Solaris 9 发行版之前的 Solaris 发行版**）

在 Solaris 8 版本中，缺省值说明如下：

Maximum of the triplet (200, `tune_t_minarmem` + 100, [10% of memory available at boot time])

常规文件系统参数

ncsize（Solaris 9 以及早期发行版）

在 Solaris 9 发行版以及早期发行版中，将何时更改说明中的 nfs:nrnode 参数错误地标识为 nfs:nfs_rnode。

UFS 可调参数

bufhwm（Solaris 9 发行版）

Solaris 10 发行版中此参数信息有了很大改变。有关更多信息，请参见第 60 页中的 "bufhwm 和 bufhwm_pct"。

说明	用于高速缓存 I/O 缓冲区的最大内存量。这些缓冲区用于写入文件系统元数据（超级块、inode、间接块和目录）。可以根据需要分配缓冲区，直到要分配的量超过 bufhwm 为止。此时，将回收足够的缓冲区以满足请求。 鉴于历史原因，此参数不需要 ufs: 前缀。
数据类型	带符号整数
缺省值	物理内存的 2%
范围	80 KB 到物理内存的 20%
单位	KB
是否为动态	否。参数值首先用于计算散列桶的大小，然后会存储到可在分配和解除分配缓冲区时调整字段值的数据结构中。在正在运行的系统上，不遵守封锁协议而直接尝试调整此值可能会导致错误操作。
验证	如果 bufhwm 小于 80 KB，或者大于物理内存的 20% 与当前内核堆大小的两倍之间的较小者，则此参数会重置为物理内存的 20% 与当前内核堆大小的两倍之间的较小者。系统控制台上和 /var/adm/messages 文件中会出现以下消息。 `"binit: bufhwm out of range (value attempted). Using N."` "value attempted" 是指在 /etc/system 中输入的值或者使用 kadb -d 命令输入的值。*N* 是系统基于可用系统内存计算所得的值。
何时更改	由于仅在需要时才会分配缓冲区，因此，缺省设置的开销是由于分配多个控制结构以处理尽可能多的缓冲区产生的。在 32 位内核上，这些结构在每个可能的缓冲区中会占用 52 字节；在 64 位内核上，在每个可能的缓冲区中会占用 104 字节。在 512 MB 的 64 位内核上，将占用 104*10144 字节（约 1 MB）。标题分配假设缓冲区大小为 1 KB，尽管在大多数情况下缓冲区大小大于该值。

通过使用内核调试程序查看内核中的 bfreelist 结构，可以找出缓冲池中尚未分配的内存量。结构中的关键字段为 bufsize，此字段是指可能的剩余内存（以字节为单位）。可以使用 mdb 通过 buf 宏查看此字段：

```
# mdb -kLoading modules: [ unix krtld genunix ip nfs ipc ]

> bfreelist$<buf

bfreelist:

[ elided ]

bfreelist + 0x78:    bufsize            [ elided ]

                           75734016
```

在内存为 6 GB 的此系统上，bufhwm 为 122277。不能直接确定所使用的标题结构数，因为请求的实际缓冲区大小通常大于 1 KB。但是，可以从此系统的控制结构分配中有利地回收一些空间。

如果 512 MB 系统上有相同结构，则表明 10144 KB 中仅有 4 KB 尚未分配。使用 kstat -n biostats 检查 biostats kstat 时，还会发现系统具有合理的 buffer_cache_hits 与 buffer_cache_lookups 的比率。这表明缺省设置对于该系统是合适的。

承诺级别　　不稳定

NFS 可调参数

nfs:nrnode (Solaris 9 8/03)

Solaris 10 说明已更新为包括文本 "NFS version 4 client"。

nfs:nfs_write_error_interval (Solaris 9 8/03)

Solaris 10 说明已更新为包括文本 "NFS version 4 client"。

nfs:nfs_write_error_to_cons_only (Solaris 9 8/03)

Solaris 10 说明已更新为包括文本 "NFS version 4 client"。

nfs:nfs_disable_rddir_cache (Solaris 9 8/03)

Solaris 10 的何时更改文本已更新为包括文本 "NFS version 4 client"。

nfs:nfs3_max_transfer_size (Solaris 9 8/03)

Solaris 10 缺省值说明已从 32, 768 (32 KB) 更新为 1,048,576 (1 MB)。

TCP/IP 可调参数

ip_forward_src_routed 和 ip6_forward_src_routed (Solaris 10)

在 Solaris 9 和 Solaris 10 发行版中，错误地记录了这些参数的缺省值。正确的缺省值已禁用。有关更多信息，请参见第 115 页中的 "ip_forward_src_routed 和 ip6_forward_src_routed"。

ip_multidata_outbound (Solaris 10)

Solaris 10 6/06 发行版中增强了此参数，以将 IP 分段成批传送到网络驱动程序。有关更多信息，请参见第 116 页中的 "ip_multidata_outbound"。

说明　使网络栈在传输过程中能够一次向网络设备驱动程序发送多个包。

启用此参数可以通过提高主机 CPU 使用率和/或网络吞吐量来减少每个包的处理开销。

多数据传输 (multidata transmit, MDT) 功能仅对支持此功能的设备驱动程序有效。

另请参见第 128 页中的 "tcp_mdt_max_pbufs"。

缺省值　1（启用）

范围　0（禁用）或 1（启用）

是否为动态　是

何时更改　如果出于调试目的或任何其他原因不希望启用此参数，请将其禁用。

承诺级别　不稳定

更改历史记录　有关信息，请参见第 160 页中的“ip_multidata_outbound (Solaris 9 8/03)”。

ip_multidata_outbound (Solaris 9 8/03)

Solaris 9 8/03 发行版之后的发行版中对此参数信息进行了显著的更改。有关更多信息，请参见第 116 页中的“ip_multidata_outbound”。

说明　使用此参数，网络栈可以在传输过程中向网络设备驱动程序一次发送多个包。

启用此参数可以通过提高主机 CPU 使用率和/或网络吞吐量，来减少每个包的处理成本。

多数据传输 (multidata transmit, MDT) 功能仅对支持此功能的设备驱动程序有效。

要使用 MDT 参数，必须在 /etc/system 文件中启用以下参数：

```
set ip:ip_use_dl_cap = 0x1
```

缺省值　禁用

范围　0（禁用）、1（启用）

是否为动态　是

何时更改　可以随时启用此功能，以便在注意以下事项的同时来提高系统性能：

- 启用此功能可能会更改 IP 层和 DLPI 提供器之间的所有包的外观。因此，使用 ifconfig 的 modinsert 功能（此功能无法识别 MDT STREAMS 数据类型）动态插入 IP 层和 DLPI 提供器之间的任何第三方 STREAMS 模块可能无法工作。

  使用 autopush(1m) 机制插入 IP 层和 DLPI 提供器之间的模块也可能无法工作。
- 如果 STREAMS 模块无法识别 MDT，请将此功能保持为禁用状态。例如，ipfilter、Checkpoint Firewall-1 之类的公共域实用程序无法识别 MDT。

承诺级别　不稳定

ip_squeue_fanout (Solaris 10)

在 Solaris 10 6/06 发行版中，修改了此参数的说明。

说明 (Solaris 10)　确定将 TCP/IP 连接与 squeue 关联的模式。

	值为 0 时会将新的 TCP/IP 连接与创建此连接的 CPU 进行关联。值 1 会将此连接与随机 CPU 进行关联，从而在系统中的所有 CPU 和所有 squeue 中有效分布负载。
说明 (Solaris 10 6/06)	确定将 TCP/IP 连接与 squeue 关联的模式。 值为 0 时会将新的 TCP/IP 连接与创建此连接的 CPU 进行关联。值为 1 时会将此连接与属于不同 CPU 的多个 squeue 进行关联。用于扇出 (fanout) 连接的 squeue 数基于 第 118 页中的 “ip_soft_rings_cnt” 。

ip_soft_rings_cnt

此参数为 Solaris 10 6/06 发行版中的新增参数。有关更多信息，请参见第 118 页中的 “ip_soft_rings_cnt” 。

ip_squeue_write（ **Solaris 10 发行版** ）

在 Solaris 10 发行版中，此参数的名称已更改为 ip_squeue_enter。有关更多信息，请参见第 117 页中的 “ip_squeue_enter” 。

tcp_conn_hash_size（ **Solaris 9 发行版** ）

在 Solaris 10 发行版中，删除了此参数。

说明	控制 TCP 模块中用于 TCP 连接的散列表的大小。
数据类型	带符号整数
缺省值	512
范围	512 到 1,073,741,824
含义	值应该是 2 的幂。
是否为动态	否。只能在引导系统时更改此参数。
验证	如果将此参数设置为不是 2 的幂的值，则会向上舍入为最接近 2 的幂的数。
何时更改	如果系统始终具有数万个 TCP 连接，请相应地增大该值。使用缺省值时，TCP 最多可以执行数千个活动连接。请注意，增加散列表大小意味着占用更多内存，因此，请设置适当的值以免造成不必要的内存浪费。
承诺级别	不稳定

tcp_wscale_always（Solaris 9 发行版）

在 Solaris 10 发行版中，已将此参数的缺省值更改为启用。有关更多信息，请参见第 121 页中的 “tcp_wscale_always”。

ipc_tcp_conn_hash_size（Solaris 9 发行版）

在 Solaris 10 发行版中，删除了此参数。

说明	控制 IP 模块中用于所有活动（处于 ESTABLISHED 状态）的 TCP 连接的散列表的大小。
数据类型	无符号整数
缺省值	512
范围	512 到 2,147,483,648
含义	此参数应该是 2 的幂。
是否为动态	否。只能在引导系统时更改此参数。
验证	如果将此参数设置为不是 2 的幂的值，则会向上舍入为最接近 2 的幂的数。
何时更改	如果系统始终具有数万个活动 TCP 连接，请相应地增大该值。使用缺省值时，系统最多可以执行数千个活动连接。请注意，增加散列表大小意味着占用更多内存，因此，请设置适当的值以免造成不必要的内存浪费。
承诺级别	不稳定

tcp_compression_enabled（Solaris 9 发行版）

在 Solaris 10 发行版中，删除了此参数。

说明	如果设置为 1，则会压缩处于 TIME-WAIT 状态的 TCP 连接的协议控制块，以减小内存使用率。如果设置为 0，则不执行压缩操作。另请参见第 125 页中的 “tcp_time_wait_interval”。
缺省值	1（启用）
范围	0（禁用）、1（启用）
是否为动态	是
何时更改	请勿禁用压缩机制。
承诺级别	不稳定

ip_forwarding 和 ip6_forwarding（Solaris 9 发行版）

在 Solaris 10 发行版中，这些参数已过时。

说明 控制 IP 在接口之间是进行 IPv4 转发还是进行 IPv6 转发。另请参见第 163 页中的"*xxx*:ip_forwarding（Solaris 9 发行版）"。

缺省值 0（禁用）

范围 0（禁用）或 1（启用）

是否为动态 是

何时更改 如果需要 IP 转发，请启用此参数。

承诺级别 不稳定

xxx:ip_forwarding（Solaris 9 发行版）

在 Solaris 10 发行版中，此参数已过时。

说明 针对特定的 *xxx* 接口启用 IPv4 转发。此参数的确切名称为 *interface-name*:ip_forwarding。例如，两个接口为 hme0 和 hme1。以下是它们对应的参数名称：

hme0:ip_forwarding 和 hme1:ip_forwarding

缺省值 0（禁用）

范围 0（禁用）或 1（启用）

是否为动态 是

何时更改 如果需要 IPv4 转发，请使用此参数针对每个接口启用转发功能。

承诺级别 不稳定

tcp_conn_req_max_q0（Solaris 8 发行版）

在更高的 Solaris 发行版中，何时更改文本已从：

何时更改 对于可能收到过多连接请求的应用程序（如 Web 服务器），可以增大缺省值以便与传入速率相匹配。

下面介绍了 tcp_conn_req_max_q0 与每个套接字的最大暂挂连接数之间的关系。

收到连接请求后，TCP 首先检查等待被接受的暂挂 TCP 连接（已执行三次握手）的数目 (*N*) 是否超过最大侦听器数。如果连接数过多，则会拒绝请求。如果连接数为允许数目，则 TCP 会检查未完成的暂挂 TCP 连接数是否超过 *N* 与 `tcp_conn_req_max_q0` 的总和。如果没有超过，则接受请求。否则，会删除最早未完成的暂挂 TCP 请求。

修改为：

何时更改　对于可能收到过多连接请求的应用程序（如 Web 服务器），可以增大缺省值以便与传入速率相匹配。

下面介绍了 `tcp_conn_req_max_q0` 与每个套接字的最大暂挂连接数之间的关系。

收到连接请求时，TCP 首先检查侦听器等待被接受的暂挂 TCP 连接（已执行三次握手）数目是否超过最大值 (*N*)。如果连接数过多，则会拒绝请求。如果连接数为允许数目，则 TCP 会检查未完成的暂挂 TCP 连接数是否超过 N 与 `tcp_conn_req_max_q0` 的总和。如果没有超过，则接受请求。否则，会删除最早未完成的暂挂 TCP 请求。

UDP 可调参数

udp_xmit_hiwat（Solaris 9 发行版）

在 Solaris 10 发行版中，更改了缺省值和值的范围。

缺省值　8192 字节

范围　4096 到 65,536 字节

udp_recv_hiwat（Solaris 9 发行版）

在 Solaris 10 发行版中，更改了缺省值和值的范围。

缺省值　8192 字节

范围　4096 到 65,536 字节

udp_max_buf（Solaris 9 发行版）

在 Solaris 10 发行版中，更改了缺省值。

缺省值　262, 144 字节

网络高速缓存和加速器 (Network Cache and Accelerator, NCA) 可调参数

sq_max_size（Solaris 9 12/02 发行版）

更高的 Solaris 发行版中对此参数信息进行了显著的更改。有关更多信息，请参见第 146 页中的 “sq_max_size”。

说明	目标流队列生成 QFULL 消息之前 syncq 的深度（消息数）。
缺省值	2 条消息
范围	1 到 0（无限制）
是否为动态	否
何时更改	当 NCA 在高内存的系统上运行时，请增大此参数以允许驱动程序对更多的数据包进行排队。如果服务器的负载过重，则增大此参数可使模块和驱动程序处理更多数据，而无需丢弃数据包或将其进行存储。
承诺级别	不稳定

过时或已删除的参数

本节介绍最新的 Solaris 发行版中过时或已删除的参数。

与分页相关的可调参数

cachefree（Solaris 8 发行版）

在 Solaris 9 以及更高发行版中已过时。

说明　Solaris 8 发行版更改了高速缓存文件系统页的方法。这些更改包含优先级分页功能。

注 – 请在 /etc/system 文件中同时删除 cachefree 和 priority_paging 设置。

高速缓存更改可以消除文件系统活动对虚拟内存造成的大部分压力。一些统计信息表明会出现以下新行为：

- 页回收率较高，因为现在可在 I/O 完成之后显式地将页添加到可用列表中。
- 可用内存现在较高，因为可用内存计数现在包括大量文件高速缓存。

- 扫描速度显著降低。

承诺级别　过时

priority_paging（Solaris 8 发行版）

在 Solaris 9 以及更高发行版中已过时。

说明　此变量可将 cachefree 设置为 lotsfree 的 2 倍。

Solaris 8 发行版更改了高速缓存文件系统页的方法。这些更改包含优先级分页功能。

注 – 请在 /etc/system 文件中同时删除 cachefree 和 priority_paging 设置。

承诺级别　过时

tune_t_gpgslo（Solaris 7 发行版）

说明　在 Solaris 8 发行版中已过时。保留此变量是出于兼容性原因。

tune_t_minasmem（Solaris 7 发行版）

说明　在 Solaris 8 发行版中已过时。保留此变量是出于兼容性原因。

System V 消息队列参数

msgsys:msginfo_msgmni（Solaris 9 发行版）

在 Solaris 10 发行版中已过时。

说明　可以创建的最大消息队列数。

数据类型　带符号整数

缺省值　50

范围　0 到 MAXINT

是否为动态　否。已装入 msginfo 结构的 msgmni 字段中。

验证　无

何时更改　当 msgget(2) 调用返回错误 ENOSPC 时或者根据软件供应商的建议进行更改。

承诺级别　不稳定

msgsys:msginfo_msgtql（Solaris 9 发行版）

在 Solaris 10 发行版中已过时。

说明	可以创建的最大消息数。如果 msgsnd 调用尝试超过此限制，则请求会被延迟直到消息标题可用为止。或者，如果请求设置了 IPC_NOWAIT 标志，则请求会失败，并且出现错误 EAGAIN。
数据类型	带符号整数
缺省值	40
范围	0 到 MAXINT
是否为动态	否。已装入 msginfo 结构的 msgtql 字段中。
验证	无
何时更改	当 msgsnd() 调用阻止或返回错误 EGAIN 时或者根据软件供应商的建议进行更改。
承诺级别	不稳定

msgsys:msginfo_msgmnb（Solaris 9 发行版）

在 Solaris 10 发行版中已过时。

说明	任一消息队列中可存在的最大字节数。
数据类型	无符号长整数
缺省值	4096
范围	0 到物理内存量
单位	字节
是否为动态	否。已装入 msginfo 结构的 msgmnb 字段中。
验证	无
何时更改	当 msgsnd() 调用阻止或返回错误 EGAIN 时或者根据软件供应商的建议进行更改。
承诺级别	不稳定

msgsys:msginfo_msgssz（Solaris 9 发行版）

在 Solaris 10 发行版中已删除。

说明	指定系统用于管理消息缓冲区空间的块大小。
数据类型	带符号整数
缺省值	40

范围　0 到 MAXINT

是否为动态　否。已装入 msginfo 结构的 msgtql 字段中。

验证　将为了支持消息和队列而创建的最大数量的数据结构所占的空间与装入模块时的可用内核内存的 25% 进行比较。如果数据结构数过大，则会拒绝装入消息队列模块，并且功能不可用。此计算包括可能由消息占用的空间。仅当首次装入模块时，才会出现这种情况。

何时更改　当缺省值不够时进行更改。通常根据软件供应商的建议进行更改。

承诺级别　过时

msgsys:msginfo_msgmap（Solaris 9 发行版）

在 Solaris 10 发行版中已删除。

说明　系统支持的消息数。

数据类型　带符号整数

缺省值　100

范围　0 到 MAXINT

是否为动态　否

验证　将为了支持消息和队列而创建的最大数量的数据结构所占的空间与装入模块时的可用内核内存的 25% 进行比较。如果数据结构数过大，则会拒绝装入消息队列模块，并且功能不可用。此计算包括可能由消息占用的空间。仅当首次装入模块时，才会出现这种情况。

何时更改　当缺省值不够时进行更改。通常根据软件供应商的建议进行更改。

承诺级别　过时

msgsys:msginfo_msgseg（Solaris 9 发行版）

在 Solaris 10 发行版中已删除。

说明　系统用作可用消息内存池的 msginfo_msgssz 段数。可用于消息的总内存为 msginfo_msgseg * msginfo_msgssz。

数据类型　带符号短整数

缺省值　1024

范围　0 到 32,767

是否为动态　否

验证　将为了支持消息和队列而创建的最大数量的数据结构所占的空间与装入模块时的可用内核内存的 25% 进行比较。如果数据结构数过大，则会拒绝装

入消息队列模块，并且功能不可用。此计算不包括可能由消息占用的空间。仅当首次装入模块时，才会出现这种情况。

何时更改　当缺省值不够时进行更改。通常根据软件供应商的建议进行更改。

承诺级别　过时

msgsys:msginfo_msgmax（**Solaris 9 发行版**）

在 Solaris 10 发行版中已删除。

说明　System V 消息的最大大小。

数据类型　无符号长整数

缺省值　2048

范围　0 到物理内存量

单位　字节

是否为动态　否。已装入 msginfo 结构的 msgmax 字段中。

验证　无

何时更改　当 msgsnd(2) 调用返回错误 EINVAL 时或者根据软件供应商的建议进行更改。

承诺级别　不稳定

System V 信号参数

semsys:seminfo_semmni（**Solaris 9 发行版**）

在 Solaris 10 发行版中已过时。

说明　指定最大信号标识符数。

数据类型　带符号整数

缺省值　10

范围　1 到 65,535

是否为动态　否

验证　与 SEMA_INDEX_MAX（当前为 65,535）进行比较，如果数值较大，则重置为该值。警告消息会写入到控制台或消息文件中，或者同时写入到两者中。

何时更改　当缺省集数不够时进行更改。通常根据软件供应商的建议进行更改。如果尝试创建的集数多于当前配置的集数，则不会显示任何错误消息。相反，应用程序会收到来自 semget 调用的返回代码 ENOSPC。

有关更多信息，请参见semget(2)。

承诺级别　不稳定

semsys:seminfo_semmsl（Solaris 9 发行版）

在 Solaris 10 发行版中已过时。

说明　指定每个信号标识符的最大 System V 信号数。

数据类型　带符号整数

缺省值　25

范围　1 到 MAXINT

是否为动态　否

验证　将信号及其支持数据结构可能占用的空间量与首次装入模块时的可用内核内存的 25% 进行比较。如果超过了内存阈值，则会拒绝装入模块，并且信号功能不可用。

何时更改　当缺省值不够时进行更改。通常根据软件供应商的建议进行更改。如果尝试在一个集中创建的信号数多于当前配置的信号数，则不会显示任何错误消息。应用程序会收到来自 semget(2) 调用的返回代码 EINVAL。

承诺级别　不稳定

semsys:seminfo_semopm（Solaris 9 发行版）

在 Solaris 10 发行版中已过时。

说明　指定每个 semop 调用的最大 System V 信号操作数。此参数是指 sops 阵列中为 semop() 系统调用提供的 sembufs 的数目。有关更多信息，请参见 semop(2)。

数据类型　带符号整数

缺省值　10

范围　1 到 MAXINT

是否为动态　否

验证　将信号及其支持数据结构可能占用的空间量与首次装入模块时的可用内核内存的 25% 进行比较。如果超过了内存阈值，则会拒绝装入模块，并且信号功能不可用。

何时更改　当缺省值不够时进行更改。通常根据软件供应商的建议进行更改。如果尝试在单个 semop 调用中执行的信号操作数多于当前允许的信号操作数，则不会显示任何错误消息。相反，应用程序会收到来自 semop() 调用的返回代码 E2BIG。

承诺级别　不稳定

semsys:seminfo_semmns（Solaris 9 发行版）

在 Solaris 10 发行版中已删除。

说明	系统上的最大 System V 信号数。
数据类型	带符号整数
缺省值	60
范围	1 到 MAXINT
是否为动态	否
验证	将信号及其支持数据结构可能占用的空间量与首次装入模块时的可用内核内存的 25% 进行比较。如果超过了内存阈值，则会拒绝装入模块，并且信号功能不可用。
何时更改	当缺省信号数不够时进行更改。通常根据软件供应商的建议进行更改。如果尝试创建的信号数多于当前配置的信号数，则不会显示任何错误消息。应用程序会收到来自 semget(2) 调用的返回代码 ENOSPC。
承诺级别	不稳定

semsys:seminfo_semmnu（Solaris 9 发行版）

在 Solaris 10 发行版中已删除。

说明	System V 信号系统支持的撤消结构总数。
数据类型	带符号整数
缺省值	30
范围	1 到 MAXINT
是否为动态	否
验证	将信号及其支持数据结构可能占用的空间量与首次装入模块时的可用内核内存的 25% 进行比较。如果超过了内存阈值，则会拒绝装入模块，并且信号功能不可用。
何时更改	当缺省值不够时进行更改。通常根据软件供应商的建议进行更改。如果尝试执行的撤消操作数多于当前配置的撤消操作数，则不会显示任何错误消息。系统用完撤消结构时，应用程序会收到来自 semop(2) 调用的返回值 ENOSPC。
承诺级别	不稳定

semsys:seminfo_semume（Solaris 9 发行版）

说明	在 Solaris 10 发行版中已删除。 可以由任一进程使用的最大 System V 信号撤消结构数。

数据类型	带符号整数
缺省值	10
范围	1到MAXINT
是否为动态	否
验证	将信号及其支持数据结构可能占用的空间量与首次装入模块时的可用内核内存的25%进行比较。如果超过了内存阈值，则会拒绝装入模块，并且信号功能不可用。
何时更改	当缺省值不够时进行更改。通常根据软件供应商的建议进行更改。如果尝试执行的撤消操作数多于当前配置的撤消操作数，则不会显示任何错误消息。应用程序会收到来自 `semop(2)` 调用的返回代码 `EINVAL`。
承诺级别	不稳定

`semsys:seminfo_semvmx`（**Solaris 9 发行版**）

在Solaris 10发行版中已删除。

说明	可以设置的最大信号值。
数据类型	无符号短整数
缺省值	32,767
范围	1到65,535
是否为动态	否
验证	无
何时更改	当缺省值不够时进行更改。通常根据软件供应商的建议进行更改。如果超过了最大值，则不会显示任何错误消息。应用程序会收到来自 `semop(2)` 调用的返回代码 `ERANGE`。
承诺级别	不稳定

`semsys:seminfo_semaem`（**Solaris 9 发行版**）

在Solaris 10发行版中已删除。

说明	可以在撤消结构中设置的最大信号值。
数据类型	无符号短整数
缺省值	16,384
范围	1到65,535
是否为动态	否
验证	无

何时更改　当缺省值不够时进行更改。通常根据软件供应商的建议进行更改。如果尝试执行的撤消操作数多于当前配置的撤消操作数，则不会显示任何错误消息。应用程序会收到来自 semop(2) 调用的返回代码 EINVAL。

承诺级别　不稳定

semsys:seminfo_semmap（Solaris 7 发行版）

在 Solaris 8 发行版中已过时。在内核中提供此变量是出于兼容性原因，但不会再使用它。

semsys:seminfo_semusz（Solaris 7 发行版）

在 Solaris 8 发行版中已过时。将忽略输入的任何值。

System V 共享内存参数

shmsys:shminfo_shmmni（Solaris 9 发行版）

在 Solaris 10 发行版中已过时。

说明　对可以创建的共享内存段数设置的系统范围限制。

数据类型　带符号整数

缺省值　100

范围　0 到 MAXINT

是否为动态　否。已装入 shminfo 结构的 shmmni 字段中。

验证　将根据装入模块时的当前可用内核内存的 25%，对支持 System V 共享内存的最大可能数量的数据结构占用的空间量进行检查。如果占用的内存过大，则尝试装入模块时会失败。

何时更改　当系统限制过低时进行更改。通常根据软件供应商的建议进行更改。

承诺级别　不稳定

shmsys:shminfo_shmmax（Solaris 9 发行版）

在 Solaris 10 发行版中已过时。

说明　可以创建的 System V 共享内存段的最大大小。此参数是指在应用程序查看其是否实际具有创建请求的内存段所需物理资源之前检查的上限。

尝试创建大小为零或大于指定值的共享内存段时将失败，并且出现 EINVAL 错误。

此参数仅指定操作系统可以接受的共享内存段大小的最大值。能否创建内存段完全取决于系统上的可用交换空间量，对于 32 位进程，还取决于要附加的内存段的进程地址空间中是否具有足够的可用空间。

数据类型　无符号长整数

缺省值　8,388,608

范围　在 32 位系统上，为 0 到 MAXUINT32；在 64 位系统上，为 0 到 MAXUINT64。

单位　字节

是否为动态　否。已装入 `shminfo` 结构的 `shmmax` 字段中。

验证　无

何时更改　当缺省值过低时进行更改。通常根据软件供应商的建议进行更改，但是，如果不需要限定共享内存段的大小，则将此参数设置为可能的最大值不会产生负面影响。

承诺级别　不稳定

`shmsys:shminfo_shmmin`（Solaris 8 发行版）

在 Solaris 9 发行版中已过时。在内核中提供此变量是出于兼容性原因，但不会再使用它。

`shmsys:shminfo_shmseg`（Solaris 8 发行版）

在 Solaris 9 发行版中已过时。在内核中提供此变量是出于兼容性原因，但不会再使用它。

NFS 模块参数

`nfs:nfs_32_time_ok` (Solaris 7)

在 Solaris 8 发行版中已过时。



附 录 B

手册修订历史记录

本部分介绍本手册的修订历史记录。

当前版本：Solaris 10 6/06 发行版

本手册的当前版本适用于 Solaris 10 6/06 发行版。

Solaris 10 6/06 发行版中的新增参数或已更改的参数

以下参数与本文档的早期版本中的参数相比，要么进行了更改，要么进行了更正。

- ip_squeue_fanout 参数已修改。有关更多信息，请参见第 118 页中的 “ip_squeue_fanout” 。
- ip_multidata_outbound 参数已得到增强。有关更多信息，请参见第 116 页中的 “ip_multidata_outbound” 。
- 在 Solaris 9 和 Solaris 10 发行版中，错误地记录了 ip_forward_src_routed 和 ip6_forward_src_routed 参数的缺省值。这两个参数的正确缺省值为禁用。有关更多信息，请参见第 115 页中的 “ip_forward_src_routed 和 ip6_forward_src_routed” 。
- ip_squeue_write 参数名称已更改为 ip_squeue_enter。有关更多信息，请参见第 117 页中的 “ip_squeue_enter” 。
- logevent_max_q_sz 参数的缺省值已从 2000 个事件更改为 5000 个事件。有关更多信息，请参见第 32 页中的 “logevent_max_q_sz” 。
- 在 Solaris 10 发行版中，错误地记录了 lwp_default_stksize 参数。对于 SPARC 系统，其缺省值为 24,576。有关更多信息，请参见第 31 页中的 “lwp_default_stksize” 。

- 在 Solaris 10 发行版中，错误地记录了 sq_max_size 参数。其缺省值为 10000 条消息。有关更多信息，请参见第 146 页中的 "sq_max_size" 。
- 在 Solaris 10 发行版中，更改了 UDP 参数的缺省值和范围。先前未记录新值。有关新值的更多信息，请参见第 132 页中的 "UDP 可调参数" 。

Solaris 10 发行版中的新增参数或已更改的参数

Solaris 内核可调参数

以下部分介绍新增的、已更改的或已过时的内核可调参数。

常规内核参数和内存参数

第 30 页中的 "default_stksize" 参数为 Solaris 10 发行版中的新增参数。

在此发行版中，更改了第 31 页中的 "lwp_default_stksize" 和第 33 页中的 "noexec_user_stack" 参数。

UFS

在 Solaris 10 发行版中，修改了以下参数：

- 第 60 页中的 "bufhwm 和 bufhwm_pct"
- 第 57 页中的 "ncsize"

常规文件系统

以下参数为 Solaris 10 发行版中新记录的参数：

- 第 65 页中的 "freebehind"
- 第 59 页中的 "segmap_percent"
- 第 65 页中的 "smallfile"

System V 消息队列

在 Solaris 10 发行版中，已删除了以下参数：

- msgsys:msginfo_msgmap
- msgsys:msginfo_msgmax
- msgsys:msginfo_msgseg
- msgsys:msginfo_msgssz

从 Solaris 10 发行版开始，以下参数已过时：

- msgsys:msginfo_msgmnb

- msgsys:msginfo_msgmni
- msgsys:msginfo_msgtql

System V 信号

在 Solaris 10 发行版中，已删除了以下参数：

- semsys:seminfo_semmaem
- semsys:seminfo_semmap
- semsys:seminfo_semmns
- semsys:seminfo_semmnu
- semsys:seminfo_semvmx
- semsys:seminfo_semume
- semsys:seminfo_semusz

System V 共享内存

从 Solaris 10 发行版中，已删除了以下参数：

- shmsys:shminfo_shmmin
- shmsys:shminfo_shmseg

以下参数已过时。

- shmsys:shminfo_shmmni
- shmsys:shminfo_shmmax

TSB 参数

以下 TSB 参数为 Solaris 10 发行版中的新增参数：

- 第 74 页中的 "tsb_alloc_hiwater_factor"
- 第 75 页中的 "default_tsb_size"
- 第 76 页中的 "enable_tsb_rss_sizing"
- 第 76 页中的 "tsb_rss_factor"

NFS 参数

所有 NFS 参数的稳定性级别已从"发展中"更改为"不稳定"。

以下 NFSv4 参数为 Solaris 10 发行版中的新增参数：

- 第 101 页中的 "nfs:nfs4_async_clusters"
- 第 99 页中的 "nfs:nfs4_bsize"
- 第 84 页中的 "nfs:nfs4_cots_timeo"
- 第 86 页中的 "nfs:nfs4_do_symlink_cache"
- 第 87 页中的 "nfs:nfs4_dynamic"

- 第 89 页中的 “nfs:nfs4_lookup_neg_cache”
- 第 91 页中的 “nfs:nfs4_max_threads”
- 第 104 页中的 “nfs:nfs4_max_transfer_size”
- 第 93 页中的 “nfs:nfs4_nra”
- 第 82 页中的 “nfs:nfs4_pathconf_disable_cache”
- 第 95 页中的 “nfs:nfs4_shrinkreaddir”

以下 NFS 参数为 Solaris 10 发行版中的新增参数或已更改的参数：

- 第 92 页中的 “nfs:nfs_nra”
- 第 92 页中的 “nfs:nfs3_nra”
- 第 95 页中的 “nfs:nfs3_shrinkreaddir”

先前错误地提供了以下 NFS 参数，现已删除：

- nfsserv:nfs_shrinkreaddir
- nfsserv:nfs3_shrinkreaddir

TCP/IP 参数

以下 IP 参数为 Solaris 10 发行版中的新增参数：

- 第 117 页中的 “ip_squeue_worker_wait”
- 第 117 页中的 “ip_squeue_enter”
- 第 118 页中的 “ip_squeue_fanout”
- 第 129 页中的 “ipcl_conn_hash_size”

以下 TCP 参数为本 Solaris 发行版中的新增参数：

- 第 127 页中的 “tcp_rst_sent_rate_enabled”
- 第 128 页中的 “tcp_rst_sent_rate”
- 第 128 页中的 “tcp_mdt_max_pbufs”

以下 TCP/IP 参数在 Solaris 10 发行版中已过时：

- tcp_conn_hash_size
- ipc_tcp_conn_hash_size
- tcp_compression_enabled
- ip_forwarding
- ip6_forwarding
- xxx_forwarding

SCTP 可调参数

以下 SCTP 参数为 Solaris 10 发行版中的新增参数：

- 第 133 页中的 “sctp_max_init_retr”
- 第 134 页中的 “sctp_pa_max_retr”

- 第 134 页中的 “sctp_pp_max_retr”
- 第 134 页中的 “sctp_cwnd_max”
- 第 135 页中的 “sctp_ipv4_ttl”
- 第 135 页中的 “sctp_heartbeat_interval”
- 第 135 页中的 “sctp_new_secret_interval”
- 第 136 页中的 “sctp_initial_mtu”
- 第 136 页中的 “sctp_deferred_ack_interval”
- 第 136 页中的 “sctp_ignore_path_mtu”
- 第 137 页中的 “sctp_initial_ssthresh”
- 第 137 页中的 “sctp_xmit_hiwat”
- 第 137 页中的 “sctp_xmit_lowat”
- 第 137 页中的 “sctp_recv_hiwat”
- 第 138 页中的 “sctp_max_buf”
- 第 138 页中的 “sctp_ipv6_hoplimit”
- 第 138 页中的 “sctp_rto_min”
- 第 139 页中的 “sctp_rto_max”
- 第 139 页中的 “sctp_rto_initial”
- 第 139 页中的 “sctp_cookie_life”
- 第 140 页中的 “sctp_max_in_streams”
- 第 140 页中的 “sctp_initial_out_streams”
- 第 140 页中的 “sctp_shutack_wait_bound”
- 第 140 页中的 “sctp_maxburst”
- 第 141 页中的 “sctp_addip_enabled”
- 第 141 页中的 “sctp_prsctp_enabled”

系统功能参数

以下系统功能为 Solaris 10 发行版中的新增功能：

- 第 150 页中的 “autofs”
- 第 150 页中的 “ftp”
- 第 151 页中的 “nfs”

在 Solaris 10 发行版中，inetd 系统功能已过时。

删除 sun4m 支持

sun4m 平台在 Solaris 10 发行版中不受支持。为了反映 sun4m 支持的删除，对以下参数进行了修改：

- max_nprocs
- maxphys
- noexec_user_stack

Solaris 9 发行版中的新增参数或已更改的参数

以下部分介绍 Solaris 9 发行版中的新增参数或已更改的参数。

ip_policy_mask

此参数为 Solaris 9 12/02 发行版中的新增参数。有关信息，请参见第 133 页中的“ip_policy_mask”。

删除 sun4d 支持

sun4d 平台在 Solaris 9 发行版中不受支持。为了反映 sun4d 支持的删除，对以下参数进行了修改：

- max_nprocs
- maxphys
- noexec_user_stack

不支持的参数或已过时的参数

不支持 priority_paging 和 cachefree

在 Solaris 9 发行版中，不支持 priority_paging 和 cachefree 可调参数。这些参数已被增强的文件系统高速缓存体系结构所代替，此体系结构实现与优先级分页类似的分页策略，但这些分页策略始终处于启用状态。尝试在 /etc/system 文件中设置这些参数时，将导致引导时警告，例如：

```
sorry, variable 'priority_paging' is not defined in the 'kernel'

sorry, variable 'cachefree' is not defined in the 'kernel'
```

已对包含 /etc/system 文件的 SUNWcsr 软件包进行了修改，以禁止包含 priority_paging 或 cachefree 可调参数。如果升级到 Solaris 9 发行版或添加 SUNWcsr 软件包，并且 /etc/system 文件包含 priority_paging 或 cachefree 参数，则会出现以下情况：

1. 如果在 /etc/system 文件中设置 priority_paging 或 cachefree 参数，则会显示以下消息：

   ```
   /etc/system has been modified since it contains references to priority

   paging tunables. Please review the changed file.
   ```

2. 在 /etc/system 文件的任何设置 priority_paging 或 cachefree 的行前面插入注释。例如，如果将 priority_paging 设置为 1，则会在包含 priority_paging 值的行前面插入以下行：

```
*NOTE: As of Solaris 9, priority paging is unnecessary and has been removed.

* Since references to priority paging-related tunables will now result in

* boot-time warnings, the assignment below has been commented out. For more

* details,see the Solaris 9 Release Notes,or the "Solaris Tunable Parameters

* Reference Manual".
```

System V 共享内存

以下参数已过时：

- shmsys:shminfo_shmmin
- shmsys:shminfo_shmseg

Solaris 8 发行版中的新增参数

logevent_max_q_sz

此参数为 Solaris 8 1/01 发行版中的新增参数。有关信息，请参见第 32 页中的“logevent_max_q_sz”。

索引

T

U

W

X

Y